UNIVERSITÉ DE STRASBOURG

ÉCOLE DOCTORALE MATHÉMATIQUES, SCIENCES DE L'INFORMATION ET DE L'INGÉNIEUR

Laboratoire ICube – UMR 7357

THÈSE présentée par :

Oana-Teodora IOVA

soutenue le : **15 décembre 2014**

pour obtenir le grade de : **Docteur de l'université de Strasbourg**

Discipline/ Spécialité : Informatique

Standards Optimization and Network Lifetime Maximization for Wireless Sensor Networks in the Internet of Things

THÈSE dirigée par :

M. NOEL Thomas	Professeur, Université de Strasbourg

RAPPORTEURS :

M. BEYLOT André-Luc	Professeur, IRIT-ENSEEIHT Toulouse
M. PICCO Gian Pietro	Professeur, Université de Trento, Italie

---

AUTRES MEMBRES DU JURY :

M. THEOLEYRE Fabrice	Chargé de recherche, Université de Strasbourg
Mme. CARNEIRO VIANA Aline	Chargé de recherche, INRIA Saclay
M. DUDA Andrzej	Professeur, INP Grenoble

UNIVERSITÉ DE STRASBOURG

École doctorale MSII

Standards Optimization and Network Lifetime Maximization for Wireless Sensor Networks in the Internet of Things

THÈSE

présentée pour l'obtention du

Doctorat de l'Université de Strasbourg

(mention informatique)

par

Oana-Teodora IOVA

Composition du jury

Directeur de thèse :	Prof. Thomas NOEL, Université de Strasbourg, France
Co-encadrant de thèse :	Dr. Fabrice THEOLEYRE, CNRS, Université de Strasbourg, France
Rapporteurs :	Prof. André-Luc BEYLOT, IRIT-ENSEEIHT Toulouse, France Prof. Gian Pietro PICCO, University of Trento, Italy
Examinateurs :	Dr. Aline CARNEIRO VIANA, INRIA Saclay, France Prof. Andrzej DUDA, INP Grenoble, France

Laboratoire des sciences de l'ingénieur, de l'informatique et de l'imagerie

ICube — UMR 7357

To my parents.

Acknowledgments

I would like to thank Dr. Fabrice Theoleyre and Prof. Thomas Noel for believing in my potential, and for guiding me during my thesis. Fabrice, I find myself lucky to have had you as my adviser. Thank you for sharing with me your knowledge, and passion for research. Thank you for always pushing me to go beyond my limits.

I would also want to thank Prof. André-Luc Beylot, Dr. Aline Carneiro Viana, Prof. Andrzej Duda, and Prof. Gian Pietro Picco for accepting to be part of my jury, for their questions, and for their useful suggestions. Your constructive comments helped me improve this manuscript.

A big thanks to all my colleagues from the Réseaux team. It was a great pleasure to share the office with my fellow PhD students Cosmin, Georgios, François, Julien B., and Damien. Thank you for your encouragements, for bearing with my complains, and for all the extra activities that made me enjoy my stay in Strasbourg. A special thanks to Erkan for his friendship, his enormous availability, and all his help. I would also want to thank Julien M., Antoine, Pascal, and Stéphane for their advices that made me a better teacher. Guillaume, Pierre and Jean-Jaques, thank you for all your help throughout these three years.

I would like to thank my *Romafia* friends for giving me the taste of research, and for continuously supporting me, even though we went separate ways. A special thanks to my two *big brothers*, Cristi and Adi, for their friendship and support. Also, thank you, Mioara and Bogdan, for your precious advices, both professional and personal. You are my role models.

I would also want to thank my girlfriends, Elena, Ioana, Tania, Roxy, Cata, Zsu, for being there for me when I needed them, for helping me surpass my problems, for sharing with me their experiences, and make me feel less alone.

I would like to thank my parents for their unconditioned support, care, and love. You always pushed me to be a better person ,and supported me in any decision. Thank you for encouraging me to follow my dreams, and for handling so well the fact that your only daughter is thousands of kilometers away from you.

I would also like to kindly thank Pierre for his love, support, and encouragement. Thank you for your precious advices, both personal and professional. Most of all, thank you for helping me keep my sanity.

Last but not least, I want to thank *Cecilia* (from PhD comics) for sharing my *joy and pain* and my passion for chocolate.

Abstract

Because of the advances of technology, the number of devices connected to the Internet now exceeds the number of people on Earth. These devices are not only computers, tablets and smartphones, but also any object that has an embedded microcontroller and a radio with an antenna. Until recent years, the Internet was an interconnection of computer networks. We are talking now about networks of computers and Things.

New protocols have been standardized in order to integrate these networks in the Internet. Among them, the IEEE 802.15.4 physical and Medium Access Control (MAC) layer protocols, and the IPv6 Routing Protocol for Low-power and Lossy Networks (RPL). This resulted in a new paradigm, called the Internet of Things.

An important part of the Internet of Things are the Wireless Sensor Networks (WSNs), which are composed of a large number of low-power devices that sense the environment and communicate this information to a central entity. The WSNs are highly unreliable, and the devices that compose them are very constrained in terms of energy, memory, and processing power.

The goal of this thesis is to improve the already standardized protocols for the Internet of Things, considering the constraints of WSNs (high loss rate, instability, low data rates, etc.), and of the devices that compose it.

First, we proposed a new MAC layer broadcast mechanism in IEEE 802.15.4, to ensure a reliable delivery of the control packets from the upper layers (especially from RPL). Then, we provided an exhaustive evaluation of RPL, and highlighted an instability problem: existing routing metrics used to build the topology never lead to network stability. These instabilities generate a large overhead, consuming a lot of energy.

Since the lifetime of WSNs is very limited, as most of their devices are energy constrained, we proposed a new routing metric that estimates the *Expected Lifetime* of a node, according to its residual energy, the link quality to its neighbors, and the current traffic conditions. By using this metric with RPL, we identify the bottlenecks of the network in terms of energy (i.e., the nodes that have the least residual energy), and we route the packets in order to maximize their lifetime.

Finally, we proposed to enhance RPL by proposing a multipath approach, in order to create energy-balanced paths. We identify the weakest nodes in the network (in terms of energy) by using the Expected Lifetime metric. Then, we forward the traffic through all the next hops, so that each path consumes the same quantity of energy, maximizing furthermore the lifetime of the network.

Publications

The contributions of this thesis have been published or submitted in several peer-reviewed international and national conferences.

Journals

- O. Iova, F. Theoleyre and T. Noel, *Using Multiparent Routing in RPL to Increase the Stability and the Lifetime of the Network*, **to appear in** Elsevier Ad Hoc Networks 2015

International Conferences

- O. Iova, F. Theoleyre and T. Noel, *Exploiting Multiple Parents in RPL to Improve both the Network Lifetime and its Stability*, **to appear in** IEEE International Conference on Communications (ICC) 2015
- O. Iova, F. Theoleyre and T. Noel, *Improving the Network Lifetime with Energy-Balancing Routing: Application to RPL*, IFIP/IEEE Wireless and Mobile Networking Conference (WMNC), Vilamoura, Portugal, May, 2014
- O. Iova, F. Theoleyre, M. Zou and J. Lu, *Efficient and Reliable MAC-Layer Broadcast for IEEE 802.15.4 Wireless Sensor Networks*, , IFIP/IEEE Wireless and Mobile Networking Conference (WMNC), Vilamoura, Portugal, May, 2014
- O. Iova, F. Theoleyre and T. Noel, *Stability and Efficiency of RPL under Realistic Conditions in Wireless Sensor Networks*, IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), London, UK, Sept. 2013

National Conferences

- O. Iova, F. Theoleyre and T. Noel, *Exploiter Plusieurs Parents avec RPL pour Améliorer la Stabilité* , Algorithms for Telecommunications (Algotel), Le-Bois-Plage-en-Ré, France, June, 2014

Table of Contents

Chapter 1

Introduction and Context

Until recent years, the Internet was seen as an interconnection of computer networks. Starting from 1969, when the Advanced Research Projects Agency Network (ARPANET) was created, the number of interconnected computers grew rapidly. By 2003, more than 500 million devices (personal computers, PDAs, etc.) were connected to the Internet [Eva11].

The advances of technology made possible not only the commercialization of smartphones and tablet PCs, but also of small, uniquely identifiable devices that have communication capabilities. These devices can be embedded in different objects, in homes or in the environment, allowing people, animals, and any kind of things to be connected to the Internet. As a result, in 2008, the number of devices connected to the Internet exceeded the number of people on Earth [Cis]. We are talking now about networks of computers and Things: the Internet of Things.

The Internet of Things includes fixed, portable or mobile devices, with constrained processing power, memory, and energy. Some of these devices, called sensor nodes, are capable of gathering sensory information (e.g., temperature, pollution, movement, etc.) and communicate among each other, forming a WSN.

WSNs have several applications from air-quality monitoring to home automation, military applications, health monitoring, etc. The view of sensor nodes slowly shifts from *fancy* gadgets to devices that improve our environment. Pollution monitoring in a city [Mao+12], parking assistance [Sma], smart metering [Tri], or water quality monitoring [O'F+07], are just a few examples of how WSNs are incorporated into our daily lives, sometimes unnoticed.

1.1 Wireless Sensor Networks

A Wireless Sensor Network (WSN) is a network composed of a large number of low-power devices that sense the environment and communicate this information to one ore more *sinks* [Aky+02a], [Ver+08]. As illustrated in Figure 1.1, communication is done through wireless links, possibly via multiple hops. The sink can act as a gateway, and connect the WSN to the Internet, or as a base station, and process the received information locally.

Inside a WSN, we can have three types of communication:

- multipoint-to-point, also called convergecast: the devices sense the environment and send the information back to the sink;
- point-to-multipoint: used by the sink to send commands to one or more devices in the network;
- point-to-point: communication between devices inside the network.

The devices composing a WSN are called *nodes*, or *motes*, and have the following characteristics:

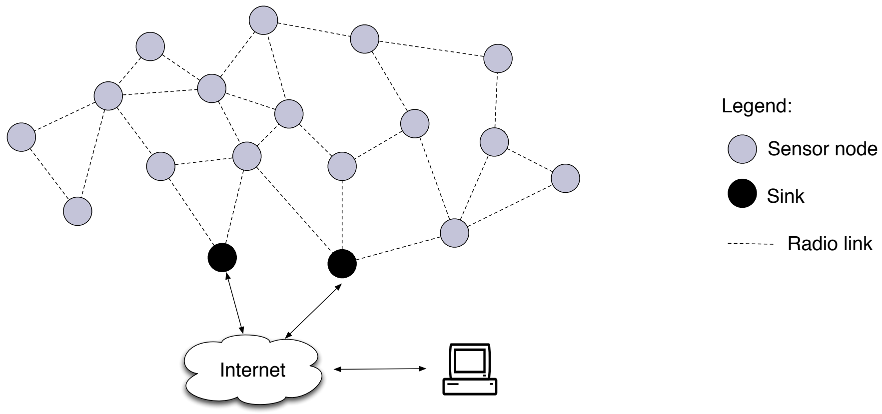


Figure 1.1: Example of a Wireless Sensor Network

- they are composed of sensing, data processing, and communication components;
- they have small sizes and they are prone to failure;
- they have limited memory, processing power, and energy (most of them are battery powered).

Because of these characteristics, the WSNs are highly unreliable, and usually densely deployed, which leads to multi-hop interference and time-varying radio link quality. This makes them part of the Low-power and Lossy Networks (LLNs) family.

1.1.1 Short History

The beginning of sensor networks can be tracked back to the U.S. Sound Surveillance System (SOSUS) in 1952. The U.S. Navy had deployed a network of hydrophones (underwater microphones) on the bottom of the ocean to detect Soviet submarines. They started with a 300m-long line array of 40 hydrophones and, after successful trials, they extended the project to the entire Est and West Coasts [Whi05]. At that moment, the information was transmitted back to the shore through multi-conductor armored cables. Nowadays, the network is used to detect earthquakes in the Pacific [CS03].

In 1980, the U.S. Defense Advanced Research Projects Agency (DARPA) started the Distributed Sensor Networks (DSN) program [Dsn]. The goal was to create a network composed of *many spatially distributed low-cost sensing nodes that collaborate with each other but operate autonomously, with information being routed to whichever node can best use the information* [CK03]. This was a very ambitions project, at a time where the sensor nodes were the size of a shoe box, and the personal computers were not existing yet. Indeed, ARPANET (the predecessor of the Internet) has just been deployed a few years earlier (in 1969), having in 1981 barely 200 hosts [CK03].

DARPA continued its implication in the development of WSNs and funded projects like the Sensor Information Technology (SensIT) [KS01] and Smart Dust [War+01]. The interest in SensIT was to create new software that can extract information from the sensors, such as detection, classification, and target tracking. Several experiments helped test and improve the software, such as the deployment of 70 nodes on a road, to track the movements of passing vehicles.

A significant breakthrough came at the end of the 1990s when Kris Pister founded the Smart Dust project with the objective of creating the smaller sensor node yet, measuring just 1 mm^3.

The resulting product of this project was a 16 mm^3 solar-powered node which has a communication range of 20 meters and a lifetime of 1 week on continuous operation. The lifetime can reach as long as 2 years, if the nodes turn their radio off 99% of the time [War+02].

Thanks to these advances, WSNs started to be seen as indispensable for the future. In 1999, the scientists predicted that millions of *embedded electronic measuring devices* will be deployed all over the planet to monitor highways, cities, the atmosphere, animals, and even our bodies [Gro99].

1.1.2 Applications of Wireless Sensor Networks

Nowadays, sensors are small, cheap and available to everyone. The technology to connect them has considerably evolved in the last 10 years. We now have networks interconnecting thousands of such devices. Moreover, we are talking about the integration of the WSNs in the emerging Internet of Things.

WSNs can be classified in function of the environment where they are deployed [YMG08]:

- terrestrial: deployed on land, they are probably the most common ones.
- underground: the sensors are deployed underground, for example in a mine [LL07], in a cave, or embedded in the road pavement.
- underwater: the sensors are immersed in water and the typical communications is done through transmission of acoustic waves [Vas+05].
- hybrid: the sensors are anchored on the sea bottom and floating within a restricted area at the surface. They are called Restricted Floating Sensors and can be used for environmental monitoring (temperature, light illuminance, sea depth, etc.) [Jia+09].

The possible applications of WSNs to the real world are substantial [Aky+02b]:

- military applications: battlefield surveillance, detection and localization of shooters [Sim+04], nuclear, biological, and chemical attack detection and reconnaissance, etc.;
- environmental applications: monitoring seismic activity [WADHW08], microclimat study of a tree [Tol+05], humidity and temperature monitoring [LBV06], forest fire detection, flood detection, wildlife monitoring, etc.;
- health applications: clinical monitoring of patients (e.g., heart rate and blood oxygenation) [Chi+10], drug administration in hospitals, etc.;
- home applications, as part of the *home automation*: light switching, open doors detection, temperature adjustment, etc.;
- urban applications: smart metering [Kan+12], traffic and parking monitoring, etc.;
- industrial applications: monitoring product quality, detection of expired products, etc;
- smart grid: the sensors are called smart meters and may use Power-Line Communication (data is carried on the same conductor that transmits the electrical power) for energy measurement, configuration, and control (e.g., [Tri]).

If we take into account the type of information that the applications need from the sensors, the WSNs can be divided into [Bar+08]:

- time driven: nodes send data periodically to the sink;
- event driven: when an event occurs (e.g., a movement has been detected) nodes forward an alert to the sink;
- query driven: nodes send data to the sink only when requested.

Table 1.1: Summary of WSN Deployments*

Name	**Year**	**No. and type of nodes**	**Type of flow**	**Communication**	**Description**
PinPtr [Sim+04]	2003	56 Mica2	event driven	convergecast multihop	sniper localization
Redwood [Tol+05]	2004	33 ~Mica2Dot	time driven 1pkt / 5 min	convergecast multihop	microclimat study
ZebraNet [Zha+04]	2004	7 custom	time driven 1 pkt / 8 min	convergecast one hop	wildlife monitoring
LOFAR-agro [LBV06]	2005	100+ ~Mica2Dot [Mic]	time driven 1pkt / 10 min	convergecast multihop	agriculture monitoring
Underwater [Vas+05]	2005	20 custom	event driven	convergecast one hop	environmental monitoring
PermaSense [Tal+07]	2006	10 TinyNode [DF+06]	time driven 1pkt / 1 min	convergecast multihop	rock monitoring
SASA [LL07]	2006	27 Mica2	time driven + event driven	convergecast + sink to nodes multihop	environmental monitoring
Lance [WADHW08]	2007	8, 25, 50 TMoteSky [Tmo]	time driven 1 pkt / 109 sec	convergecast multihop	vulcano monitoring
SensorScope [Bar+08]	2007	6-97 TinyNode	time driven 1pkt / 60 min	convergecast multihop	environmental monitoring
GreenOrbs [Mo+09]	2008	1000+ TelosB [PSC05]	time driven 1pkt / 5 min	convergecast multihop	ecological monitoring
OceanSense [Jia+09]	2008	20 TelosB	time driven N.A.	convergecast multihop	environmental monitoring
Torre Aquila [Cer+09]	2008	17 custom TMote	time driven	convergecast multihop	building monitoring
Badger [Dyo+10]	2009	24 TMoteSky	time driven 1 pkt / 15 min	convergecast multihop	wildlife monitoring
Clinical [Chi+10]	2009	59 TelosB	time driven 1pkt / 1 min	convergecast multihop	clinical monitoring
IPMS [Kim+10]	2010	~450 custom	time driven N.A.	convergecast one hop	parking monitoring
CitySee [Mao+12]	2011	1000+ TelosB	time driven 1pkt / 10 min	convergecast multihop	CO2 monitoring
Genesi [Col+13]	2011	32 CM500 [Cm5]	time driven 1 pkt / 60 min	convergecast multihop	tunnel monitoring

*N.A. – information not available;

However, considering the deployments of WSNs in the last 10 years (c.f. Table 1.1), we can notice that the most common ones are the time driven applications. Even applications such as the intrusion detection tend to use a periodic traffic, to inform the sink that the nodes are still functional. They will be sending an empty packet when no event occurs.

We can also notice that most of the WSNs are environmental applications. However, a lot of WSNs are deployed and used on a daily basis in urban environments, e.g., parking systems like SmartGrains [Sma] and SFpark [Sfp]. However, these are industrial deployments that offer ready-to-use solutions and the data is not publicly available.

No matter the classification, WSNs are an active part of our daily life. The boom of the Internet of Things paradigm made them world known and needed. They have been expected and planned since the 20th century, but the technology to develop them on a large scale only starts to catch up now. We can only think they will extend furthermore in the future.

1.2 Motivation and Assumptions

Most of the WSN deployments use custom protocols, developed for the specific needs of each application, e.g., ZebraNet [Zha+04], Torre Aquila [Cer+09], etc. These were some pioneering pieces of work that showed the relevance of WSNs. However, nowadays, we talk about the Internet of Things, and connecting all the devices to the Internet. We need standard protocols that ensure the interoperability between all these heterogenous devices.

A lot of effort have been put by the Internet Engineering Task Force (IETF) and the Institute of Electrical and Electronics Engineers (IEEE) into standardizing the protocols for the Internet of Things, and thus, for WSNs. The results were, among others, the standardization of the IEEE 802.15.4 protocol (for the physical and MAC layer), RPL (for the network layer), and more recently, the CoAP (for the application layer).

The goal of this thesis consists in solving some key problems of these standard protocols, taking into account the specific needs of the WSNs. Since the communication between devices is the building block of any network, we turned our attention to the MAC and network layers.

Considering the characteristics of the deployments in the last 10 years (Table 1.1), we decided to focus on the multihop networks, with a convergecast type of traffic, since they are the most widespread. For the same reason, we also considered only time driven applications.

1.3 Contributions

The goal of this thesis is to propose energy efficient solutions that will maximize the network lifetime, without compromising its performance. All the algorithms and mechanism that we proposed follow this goal by reducing the overhead due to control packets, keeping a low number of retransmissions, and constructing energy-balanced paths.

First, we propose a new method for the MAC layer broadcast in* IEEE 802.15.4*, to ensure a reliable delivery of the control packets from the upper layers (especially from RPL).

The synchronous mode of the IEEE 802.15.4 divides the time into active parts (when nodes exchange packets) and inactive parts (when the nodes turn off their radio to save energy). To ensure the communication between two nodes, beacons are used to synchronize their active periods. In order to efficiently deliver both beacon and broadcast packets, we propose a new mechanism for the transmission of beacons: the CBOP. To make the MAC-layer broadcast delivery more reliable, we also introduce a broadcast sequence number.

Second, we highlight the instability of RPL: whatever routing metric is used for the construction of the topology, a node changes frequently its next hop. These oscillations generate a large overhead, consuming a lot of energy.

Each routing metric is different in terms of environmental information that it uses, computational complexity, and induced overhead. Depending on the metric that RPL uses to construct the routing topology, the network will perform differently in terms of end-to-end packet delivery ratio, delay, energy consumption, or topology dynamics. We offer here a thorough evaluation of the behavior of RPL when using different routing metrics for the construction of the network topology. We also show that none of these routing metric succeeds in guaranteeing both stability and efficiency.

Third, we propose a new routing metric that maximizes the lifetime of the network.

The lifetime of WSNs is very limited, as most of their devices are energy constrained. We propose a new routing metric that estimates the *Expected Lifetime* of a node, according to its residual energy, the link quality to its neighbors, and the current traffic conditions. We take a novel approach by maximizing the lifetime of the most constrained node (i.e., maximizing the minimum lifetime rather than minimizing the average energy consumption). We identify the bottlenecks of the network in terms of energy (i.e., the nodes that have the least residual energy), and we route the packets in order to maximize their lifetime. We obtain consequently energy balanced paths, and thus, we prolong the lifetime of the network.

Finally, we propose a multiparent version of RPL, and we combine this approach with the* Expected Lifetime *routing metric, to further maximize the network lifetime.
With RPL, a node selects one preferred parent to construct a Directed Acyclic Graph (a graph without cycles). However, only this preferred parent is used for routing: the other ones have just a backup purpose. We propose to enhance RPL by exploiting several parents, in order to create energy-balanced paths. We identify the weakest nodes in the network (in terms of energy) by using the Expected Lifetime metric, and we construct accordingly the routing topology. Then, we probabilistically forward the traffic through all the parents, so that each path consumes the same quantity of energy.

1.4 Structure of the Thesis

This manuscript is organized in seven chapters. The first chapter presents an introduction to Wireless Sensor Networks, as well as the motivation and contributions of this thesis. The next chapter presents an overview of the WSN architecture, with a focus on the MAC and network layers. It also gives the reader the necessary elements for understanding the rest of this manuscript.

Starting with Chapter 3, each of the chapters presents one of the contributions of this thesis. We begin with the proposal of a new MAC layer broadcast for IEEE 802.15.4 in Chapter 3, and we continue with the study of the the dynamics of RPL in Chapter 4.

In Chapter 5 we present the *Expected Lifetime*, a new routing metric that maximizes the lifetime of the network by identifying the most energy constraint nodes in terms of energy. Chapter 6 uses this routing metric and combines it with multiparent routing in order to further maximize the network lifetime.

Finally, Chapter 7 concludes this thesis by presenting concluding remarks and opening up some perspectives.

Chapter 2

The Architecture of Wireless Sensor Networks

One of the first protocol stacks for WSNs was envisioned by Akyildiz *et al.* (Figure 2.1b) in 2001 [Aky+02b]. This was a modification of the OSI model (Figure 2.1a), where the session and presentation layers were left aside, as they were no considered useful in this environment.

After years of study, researchers realized that because of the specific constraints of WSNs, this model was not very accurate. When trying to optimize for energy consumption and high reliability with constrained devices, it is difficult to maintain the separation between layers. Cross layer solutions proved to be more efficient [RI04].

The IEEE and IETF working groups have standardized several protocols in an effort to create a protocol stack for the Low-power and Lossy Networks (LLNs), and hence, for WSNs. Since they advocate for the separation between layers, several adaptation layers had to be created, for the different standardized protocols to efficiently work together.

For example, the IEEE 802.15.4 standard has a packet size limit of 127 bytes. In order to have IPv6 connectivity over these networks, the 6LoWPAN [Mon+07] adaptation layer had to be created. Compression algorithms like the LOWPAN_IPHC Encoding [HT11] can compress the IPv6 header into 2 bytes. Another example is the 6TOP sublayer [WVW14], developed by the IETF 6TiSCH working group, which allows interactions between the link layer and the upper layers in the stack.

As we can see in Figure 2.1c, these two additional protocols (6LoWPAN and 6TOP) have been added to the WSN stack proposed for standardization.

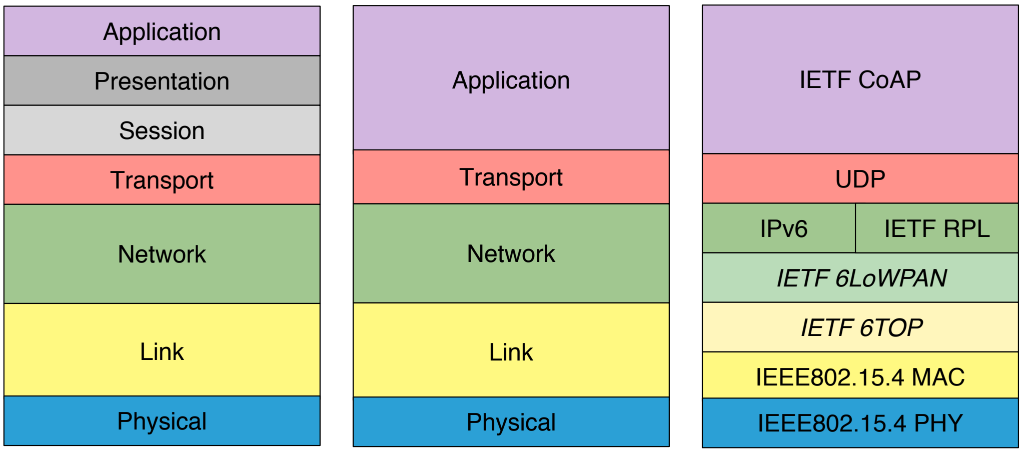
Let us present next into more details each of these layers, with a focus on the standardized protocols.

2.1 The Physical Layer

The physical layer is represented in WSNs by the radio chip. It transmits and receives bytes of data over the air, as an electromagnetic signal. Because of its fundamental role, it was the first one to be standardized.

Even though the focus of this thesis is not on the physical layer, we provide next a short *network researcher* point of view.

Baccour *et al.* presented a comprehensive study characterizing the low-power radio links [Bac+12]. Although some of these characteristics are similar to those of traditional wireless networks, the low-power radios used by these devices make the radio links even more unreliable. The main observations they made are:



(a) OSI Stack (b) WSN Stack (2001) (Reproduced from [Aky+02b]) (c) Standardized Stack (Reproduced from [AP14])

Figure 2.1: WSN Protocol Stacks

- radio links can be asymmetrical, i.e., there is a difference between the Packet Reception Ratio (PRR) of the uplink and the PRR of the downlink;
- the transmission range can be divided into three regions:
  - connected: this region contains good, stable, and mostly symmetrical radio links;
  - transitional: on the long term, the link quality varies a lot, and most of the links present asymmetry;
  - disconnected: the radio links have poor quality, and most of the times they are not used for communication.
- link quality is not correlated with distance, e.g., a node situated far away from a source may have a PRR higher than another node situated closer;
- links with very low or very high average PRR are more stable than links with intermediate values of average PRR;
- changes in the environment (e.g., human presence, obstacles, interference, etc.) may determine temporal variation of the link quality;
- concurrent transmissions and cross-channel interference have both a significant impact on the quality of the radio links;
- since WSNs operate on unlicensed Industrial, Scientific and Medical radio bands (ISM), they may suffer from external interference. For example, the 2.4 GHz frequency can be shared between WSNs, WiFi, and Bluetooth devices.

The most prominent standard in low-power radio technologies is the IEEE 802.15.4 protocol. It defines the specifications for the physical layer and the Medium Access Control sub-layer for LLNs. Being firstly defined in 2003, it was revised in 2006, and for the physical layer in 2007.

IEEE 802.15.4 also defines two measurements that can be used by the upper layers (the network or the application layers) to e.g., favor certain paths, or to help make the channel selection:

- the Link Quality Indicator (LQI): measured as *the strength and/or quality of the received packets* [802].

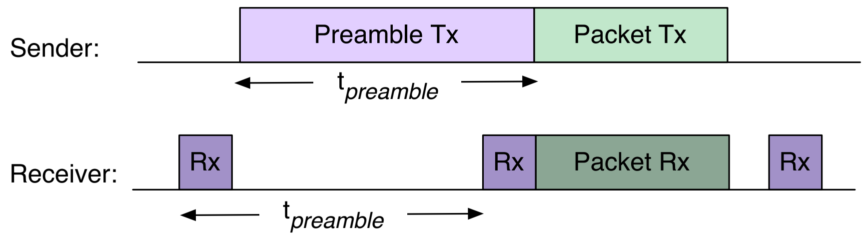


Figure 2.2: MAC protocol with preamble sampling

- the Energy Detection (ED): measured as *an estimation of the received signal power within the bandwidth of the channel* [802].

Examples of radios that are IEEE 802.15.4-compliant, are: Atmel AT86RF231, Texas Instruments CC2520, Dust Networks/Linear Technology LTC5800, Silicon Labs EM357.

2.2 The Link Layer

The Medium Access Control (MAC) of the link layer is responsible for deciding which devices have access to the wireless medium, when, and on which channel. Because of the broadcast nature of the medium, it has to deal with interference and collisions.

In WSNs, energy consumption is the primary concern, as the devices are battery powered and they have to last for several years [Bac+10b]. Between all the components of a node (radio, CPU and sensors), the most energy consuming one is the radio, i.e., transmitting and receiving messages. Compared to other types of networks, here, a MAC protocol also has to deal with energy constraints. It divides the time into duty cycles, i.e., it alternates periods of activity (transmitting and receiving data) with periods of sleep (turning the radio off). Two nodes are able to communicate with each other only when they are both in active mode at the same time. A protocol offering 1% of duty cycle means that the devices are in active mode only 1% of the time.

MAC protocols for WSNs can be divided in function of the traffic pattern into: preamble sampling protocols, protocols with common active periods, and scheduled protocols [Bac+10b].

2.2.1 Preamble Sampling Protocols

Preamble sampling protocols send one long preamble before sending a data packet. To be effective, the duration of the preamble needs to be longer than the sleeping period of the devices. As we can see in Figure 2.2, the neighboring nodes have to periodically wake up and check the medium for possible transmissions. If a node senses the medium busy, it will keep the radio on to receive the data frame. To be more energy efficient, the preamble can be divided into multiple short ones.

These protocols do not need synchronization between the nodes. Moreover, it makes it easy to add nodes after the network has been deployed. Because the preambling is energy-consuming, they are more adequate for applications with low traffic, or event-driven (e.g., alarms). Example of such protocols are: WiseMAC [EHD04], B-MAC (Berkley MAC) [PHC04], X-MAC [Bue+06], ContikiMAC [Dun11].

2.2.2 Protocols with Common Active Periods

For applications that deal with periodic traffic (e.g., monitoring), it is more suitable to use protocols with common active periods. These protocols use control packets to synchronize the

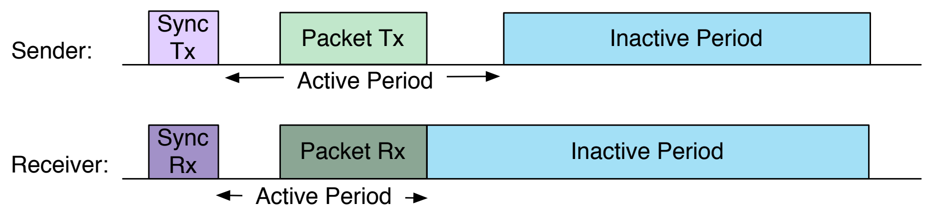


Figure 2.3: MAC protocol with common active periods

active/sleep periods of the nodes. Usually, at the beginning of each active period, a node broadcasts a beacon to announce the neighbors to synchronize with it. During the active period, they use a mechanism such as Carrier Sense Multiple Access (CSMA) to send the data packets. As we can see in Figure 2.3, after a node receives the data packet, it can directly turn off its radio, in order to save energy.

One drawback of these protocols is that setting the optimum duration of the active period is not easy. A longer active period decreases the collision probability, but also decreases the sleeping period, consuming more energy, and vice-versa. The beacon-enabled mode of IEEE 802.15.4 [802] (which we will detail in the next chapter) uses this type of synchronization. Other protocols that use common active periods are: SMAC (Sensor MAC) [YHE02], TMAC (Timeout MAC) [DL03], SWMAC (Separate Wakeup MAC) [PSH06].

2.2.3 Scheduled Protocols

Scheduled protocols use control packets to ensure that the nodes are allocated a time slot through a Time Division Multiple Access (TDMA) mechanism, and maybe even a frequency channel if Frequency Division Multiple Access (FDMA) is used. During a time slot, a node can sleep, transmit or receive a packet. Even though control packets induce more overhead in the network, this can become insignificant when the traffic is more frequent.

The advantage of these protocols is that once the schedule is set, there will be no collisions in the network. However, it is quite difficult to determine a collision-free schedule for very large networks. Example of such protocols are PEDAMACS [EV06], SMACS [Soh+00], TRAMA [ROGLA06]. The Timeslotted Channel Hopping (TSCH) mode of the IEEE 802.15.4, which is part of the standardized network stack, is also a scheduled protocol that uses both TDMA and FDMA techniques.

2.2.4 Standardized Protocols

Currently, the only MAC protocol standardized for LLNs is the IEEE 802.15.4. The IEEE 802.11ah Task Group is making efforts to standardize an amendment of the IEEE 802.11 protocol, and one of its goals is to answer the specific needs of the Wireless Sensor Networks. The standard is planned to appear in 2016 [Iee], so we will not focus here on it.

Let us thus describe into more details the IEEE 802.15.4 MAC protocol.

2.3 IEEE 802.15.4 - The Standard MAC Layer for Wireless Sensor Networks

IEEE 802.15.4 was introduced at first in Personal Area Networks (PANs) and represents now one of the most used standards in WSNs. A first version of the standard was published in 2003 and was followed by a revision in 2006. Several working groups strived to make further improvements to these standards. Even though an amendment of the standard was made in 2012, we will focus

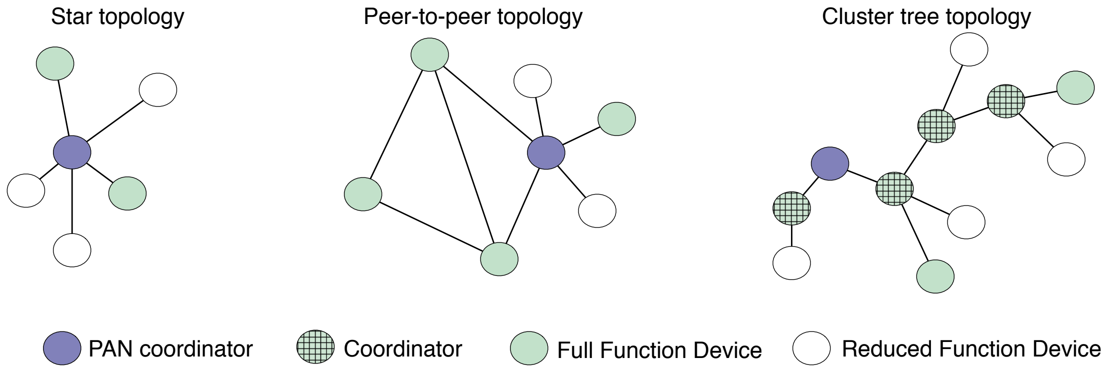


Figure 2.4: IEEE 802.15.4 topology

here on the 2006 version, since this was the newest version available at the beginning of this thesis.

We will present here the key aspects of the protocol.

2.3.1 Link Layer Topology

An IEEE 802.15.4 network is formed of several devices and one principal controller (the PAN coordinator). The devices can be either Full Function Devices (FFDs) or Reduce Function Devices (RFDs). While the FFDs support all the network functionalities, the RFDs have limited capabilities (e.g., they can sense the environment and send the information to the next hop, but they cannot act as routers and relay data packet themselves from other neighbors).

The standard defines two types of network topologies: star and peer to peer. As we can see in Figure 2.4, in a star topology, all the devices communicate only with the PAN coordinator. Usually, the devices are battery powered and only the PAN coordinator is mains powered.

In a peer-to-peer topology (Figure 2.4), the communication is not restricted to the PAN coordinator, devices being allowed to communicate with each other. In contrast to the star topology, this type of structure allows multihop communication, a key feature in WSNs. However, a node must always stay awake, since it may receive a packet from any neighbor.

A special case of the peer-to-peer topology is the cluster tree (Figure 2.4). This structure is preferred for multihop networks when a synchronization among the nodes is needed. The PAN coordinator is the root of the cluster tree, and bootstraps the network construction. Any associated node capable of routing packets, namely a FFD, may act as a *coordinator*. It forwards packets to, or from the PAN coordinator. The RFDs can attach to the cluster tree only as leaves. To improve the WSNs robustness, a cluster-Directed Acyclic Graph (directed graph with no cycles) may be used to create redundant paths [PTD11].

Since in WSNs the convergecast traffic is the most common scenario, we will refer to the PAN coordinator in the rest of this paper also as the sink of the network.

2.3.2 Medium Access

In IEEE 802.15.4-2006, the version used in this thesis, the medium access may be either asynchronous (beacon-less) or synchronous (beacon-enabled).

IEEE 802.15.4-2006: Asynchronous Mode

In this mode, nodes use an unslotted Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism, without RTS/CTS, to transmit their packets. Except for the star topology, a node must always stay awake, since it may receive a packet from any neighbor.

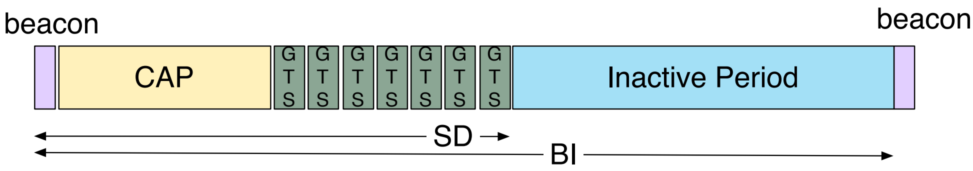


Figure 2.5: Superframe structure for IEEE 802.15.4 synchronous mode

A preamble technique may be coupled with this mode to allow a node to wake up only periodically, to capture the preamble. However, this breaks the compliance with the standard. Besides, such technique is expensive for the transmitter, and accurate only for non predictive and infrequent traffic.

IEEE 802.15.4-2006: Synchronous Mode

In the beacon-enabled mode, IEEE 802.15.4 introduces the concept of superframes (Figure 2.5). Each coordinator periodically sends – every Beacon Interval (BI) – a `beacon`, piggybacking the control information. Then, the transmissions from its children take place using a slotted CSMA/CA solution during the first part of the superframe (CAP), and with dedicated timeslots (GTS) in the second part. A Guaranteed Time Slot (GTS) has to be reserved a priori by a child with a request transmitted during the Contention Access Period (CAP). The whole active part of the superframe lasts for a Superframe Duration (SD). When a node has finished participating to the superframe, it may sleep until the next `beacon` reception/transmission.

In the beacon enabled mode, a node participates in two superframes: as a child for the superframe of its parent (designated as *outgoing*), and as a coordinator for its own superframe (designated as *incoming*). The standard specifies that both superframes are interspaced by a constant `StartTime`.

The `Superframe Duration` respectively, the `Beacon Interval`, are defined through the Superframe Order (SO), and respectively the Beacon Order (BO) values, according to the following relations:

$$\text{SD} = aBaseSuperFrameDuration * 2^{SO} \tag{2.1}$$

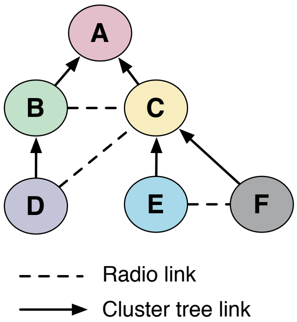
$$\text{BI} = aBaseSuperFrameDuration * 2^{BO} \tag{2.2}$$

By adjusting the BO and SO values, we can obtain a tradeoff between network capacity and energy savings. Indeed, the duty-cycle can be computed as $2^{-(BO-SO)}$. For instance, a duty cycle of 1% can be obtained if $BO - SO = 7$.

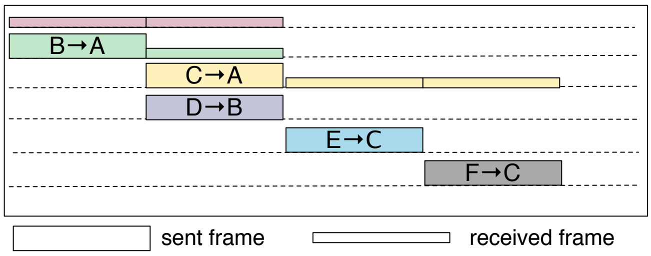
A `beacon` contains control information such as the BO and SO values, pending addresses, or the descriptor for the dedicated timeslots. If the `beacons` are lost, this can lead to desynchronization among the nodes. Moreover, some nodes will not be able to associate at all to a coordinator and hence, to the topology.

We can note that the synchronous mode enables energy savings in multihop topologies only with a cluster-tree. Indeed, with the peer-to-peer topology, a node may receive a packet from any neighbor, preventing it to sleep. In a cluster-tree, however, two nodes have to be associated with each other in order to communicate. For example, in Figure 2.6, even if node E is in the communication range of node F, they cannot directly exchange packets.

Also, a schedule tells each node when it has to wake up. As we can see in Figure 2.6, we can schedule several nodes to transmit at the same time, if they are not interfering with each other. Indeed, node C can transmit a packet to node A at the same time as D transmits to B. Once the transmission has finished, the nodes can go back to sleeping mode, saving this way energy. However, the active periods must be carefully scheduled to avoid collisions among both



(a) Cluster tree topology



(b) IEEE 802.15.4 beacon-enabled scheduling

Figure 2.6: An IEEE 802.15.4 network and its schedule

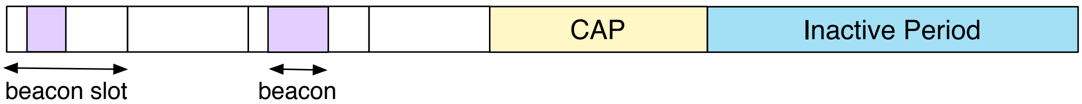


Figure 2.7: Beacon Only Period (BOP) with 4 slots shared between 2 coordinators

`beacons` and `data` packets. In particular, if `StartTime` is a constant, the `beacons` of siblings will collide, since they are sent without a CSMA/CA mechanism. This will significantly decrease the performance of the network.

There exists two main approches to reduce the number of collisions. In the Beacon Only Period (BOP), nodes rely on a TDMA approach to send their `beacons`: at the beginning of each superframe a few slots are dedicated to `beacons` [KCA07]. We can see in Figure 2.7 two coordinators sharing the same superframe, using the BOP method. Each of them chooses one of the 4 free slots to send its `beacon`. However, data packets can still collide during the Contention Access Period [Kou+06].

Wong proposed to send the `beacon` during a slot inside the CAP [Won12]. Since the data packets and the `beacons` are transmitted in the same frame, authors propose to delay the transmission of the data packets if their slot is the same. However, this can lead to collisions between both `beacons` and data packets.

A second solution relates to a variable *StartTime*: two nodes that have the same parent should not use the same *StartTime* so that their superframes will not overlap. When superframes overlap, collisions between the data packets increase, since they send data packets in the same time frame (i.e., they have the same Contention Access Period). Finding the adequate *StartTime* for all of them is equivalent to scheduling the superframes with a TDMA approach. Several distributed solutions exist (e.g., [Lee+11]). However, bandwidth is wasted since the coordinators that do not have any children will waste a *slot* for their superframe. Thus, BOP and superframe scheduling may be implemented together to reduce both collisions and the waste of bandwidth [PHT12].

2.3.3 Broadcast in IEEE 802.15.4

When it comes to how MAC-layer broadcast should be handled, IEEE 802.15.4 makes just two specifications:

1. *any frame that is broadcast shall be sent with its Acknowledgment Request subfield set to zero* [802] else, all the acknowledgements would logically collide;

2. a broadcast message *shall be transmitted immediately following the beacon with the CSMA/CA algorithm* [802].

However, the standard does not explain *what a node should do* when it must broadcast a packet to all its neighbors. Since a child may sleep immediately after having received the `beacon`, the coordinator cannot send a broadcast packet during the Contention Access Period: some nodes will be deaf and will not receive the packets. All the children must consequently stay awake during the whole active part of the superframe of its parent.

Bachir *et al.* make a distinction between neighboring broadcast (the packet has to be delivered to all the neighbors) and discovering broadcast (the packet aims at discovering new neighbors) [Bac+10a]. In multihop networks, we must implement both types of MAC-layer broadcast. Indeed, we may require discovering broadcast (e.g., for the cluster-tree reconfiguration) and neighboring broadcast (e.g., for control traffic generated by the routing protocol).

However, most of the existing proposals focus rather on network-wide broadcasting (flooding). For example, Ding *et al.* limit the transmission redundancy in Zigbee by constructing a tree [Din+06]. However, such a solution only works with the beacon-less mode of IEEE 802.15.4 and cannot be applied to low duty-cycle protocols.

In the beacon-enabled mode, the broadcast mechanism must cope both with unreliable links and with a duty-cycle MAC. Consequently, most existing solutions implement MAC-layer broadcast by duplicating the packets: a coordinator duplicates the broadcast packet into several unicast packets for each of its children and parent. Guo *et al.* reduce the network-wide broadcast delay by forwarding packets along non-optimal links of the flooding tree when the delay gain is appreciable [Guo+09]. However, this scheme relies on unicast transmissions and focuses only on the flooding case.

To reduce the flooding delay in low duty-cycle networks, Wang *et al.* proposed a forwarding selection algorithm [WL12]. The algorithm uses a mix of MAC-layer unicast and broadcast to transmit the packets to the selected forwarders. However, the authors assume the sleeping schedule is known by the transmitter, and that a MAC primitive permits to *cover* all these non-sleeping nodes, reliably.

2.4 The Network Layer

The network layer is responsible for constructing paths between the nodes in the network, and routing the data packets from the source to the destination.

Routing in WSNs has been extensively studied in the past [AY05]. The routing protocols can be categorized into geographical, proactive (table-driven), and reactive (on demand) protocols. The geographical routing protocols construct the route between two devices by using the geographical information of the nodes. The reactive protocols aim at reducing the control packets when a very low traffic has to be forwarded in the network. On the contrary, proactive protocols aim at constructing a priori efficient routes.

Geographical Routing Protocols

Geographical routing protocols use the position of the nodes in the network to route packets. The position can be given by a Global Positioning System (GPS). Each node has knowledge of the position of its neighbors and of the sink, and hence, can compute the next hop. However, any network changes are difficult to detect, since this would require an update of the geographic information both in the affected node, and in the entire network [Wat+11].

The Greedy Perimeter Stateless Routing (GPSR) [KK00] is a well known geographical routing protocol. The nodes periodically broadcast control packet in the network, which contain only their ID (e.g., IP addreess) and position. To minimize the overhead, this information may be piggybacked on the data packets.

However, GPS is highly energy consuming, and thus, it should be used carefully with constrained devices.

Reactive Routing Protocols

Reactive protocols create a route between a source and a destination, only when needed, i.e., *on demand*. These protocols are preferred when there is a low traffic in the network, but not when the applications need a low latency (because of the time it takes to create the route).

Flooding [LK01] is one of the most simple techniques that can be used for routing in WSNs: when a node has a data packet to send, it broadcasts it to all its neighbors. Then, all the neighbors will broadcast at their turn the data packet, and so on, until it reaches the destination. This algorithm does not need any topology maintenance, since it does not construct any routing topology, but it is highly energy consuming.

Different techniques have been proposed to reduce the set of neighbors that will re-transmit the data packets. For example, in the Dynamic Random Flooding (DRF) algorithm [HEB08], a node probabilistically forwards the data packet, in function of the number of hops it has already made.

Another possibility is to flood control packets before sending the data. The 6LoWPAN Ad Hoc On-Demand Distance Vector (LOAD) Routing protocol [Kim+07] is an adaptation of the Ad hoc On-Demand Distance Vector (AODV) [PBRD03] for LLNs. When a node has a packet to send, it floods the network with *route request* control packets. Each node registers locally an entry for the originator of the flooding, creating this way a reverse path. The destination replies (in unicast) with a *route reply* message. In order to reduce the size of the control messages, LOAD does not use the destination sequence number, like AODV. To ensure loop freeness, LOAD disallows the intermediate nodes to respond with a *route reply*, if they already have a path towards the destination. Finally, a bi-directional route is set between the source and the destination.

The Lightweight On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng) [Cla+14] continues the work of LOAD, in an effort to standardize a reactive routing protocol for low-power networks. Additionally, it supports an optimized flooding, and different routing metrics.

Proactive Routing Protocols

Proactive protocols create and maintain an up-to-date list of destinations and their corresponding routing tables. Contrarily to the reactive protocols, when a node needs to send a data packet, the routing path already exists. However, the proposed solution must be careful to limit the overhead in the network.

Most of the current proactive solutions in WSN are based on gradient-routing, since it is optimized for the convergecast traffic pattern [FPH05]. The Gradient Based Routing (GBR) was first introduced in 2001 [SS01]. During the control packet exchange, each node records its *height* i.e., the number of hops from the sink. The difference between the height of a node and that of its neighbor is considered the gradient on that link. Data packets are then forwarded to the neighbor having the smallest height.

The Collection Tree Protocol (CTP) [Gna+09] extends the idea from GBR by introducing the link quality estimation in the computation of the gradient. It also adds a dynamic algorithm for disseminating the control packets (i.e., the Trickle timer [Dev+11]), and a path validation technique. Even though it only supports the convergecast type of traffic, CTP represents the main routing protocol in TinyOS [Tin].

Inspired from CTP, the IETF ROLL working group standardized the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [Win+12] to offer IPv6 connectivity over networks containing up to thousand of devices. In contrast to CTP, RPL supports also multipoint-to-point traffic (from the sink towards the nodes), and point-to-point (communication among the nodes). Being part of the standardized network stack, we will detail it further more in the next section.

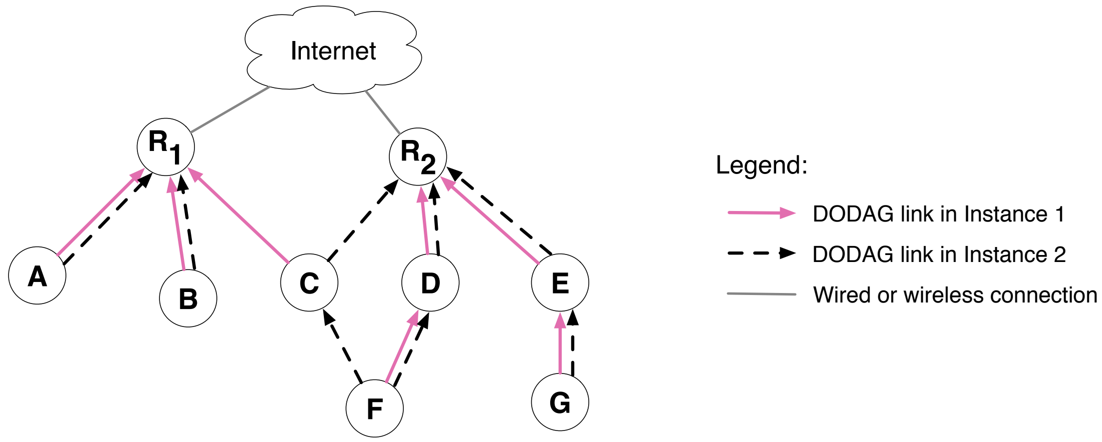


Figure 2.8: A RPL network with 2 DODAGs and 2 instances

2.5 RPL - Standard IPv6 Routing Protocol for Wireless Sensor Networks

RPL is a distance vector protocol defined for constrained devices that typically have limited processing power, memory, and energy [Win+12]. The radio links connecting them are highly unreliable, have low data rates, and their quality varies through time. The protocol was designed especially for convergecast type of traffic, but it also supports point-to-multipoint communication (from the border router toward the nodes) and point-to-point (between devices). All these characteristics make RPL a suitable routing protocol for WSNs.

The protocol was designed to operate on a variety of link layer technologies, such as low-power wireless or Power-Line Communication (PLC). It has even been enhanced to be used for Vehicular Ad Hoc Networks (VANETs) [Tia+13]. However, it requires for a radio link to be bidirectional. Since WSNs are known for having asymmetric links, RPL expects an external mechanism to validate the links that will be used for routing, before they are chosen. Examples of such mechanisms are the Neighbor Unreachability Detection (NUD) [Nar+07] and the Bidirectional Forwarding Detection (BFD) [KW10].

RPL was designed to meet the specifications of four main types of applications:

- home automation, such as light switching, controlling window shades, alarm systems, temperature monitoring, etc. [BBP10];
- building automation: building management systems deployed in different environments like universities, hospitals, office buildings, etc. [Mar+10];
- industrial applications for improving the productivity and safety of the plants [Pis+09];
- urban applications, from smart grids to environmental monitoring [Doh+09].

In order to support different applications, the network can run multiple RPL instances. Each instance is optimized for the specific constraints of the application, e.g., minimum latency, high reliability, etc. Since RPL was optimized for the convergecast traffic, it constructs a Destination-Oriented Directed Acyclic Graph (DODAG) rooted at the border router (i.e., sink), which can act as gateway to the Internet. A RPL instance can be composed of several DODAGs, one DODAG per sink. However, in a specific instance, a node can belong to only one DODAG. As we can see in Figure 2.8, the node C is connected to the sink R_1 in Instance 1, and to R_2 in Instance 2.

For the convergecast traffic, RPL uses the *upward* routes created by the DODAG. For the *downward* routes (used for communication between devices, and from the border router toward the nodes), RPL proposes two Modes of Operation (MOP):

- **storing**: each node stores downward routing tables for their sub-DODAG. At each hop on the route, the node examines its routing table and decides to which neighbor to send it next. In a point-to-point communication, the packet will travel up until the first common ancestor, and then down to the destination.

- **non-storing**: the only node storing a downward routing table is the border router, which uses source routing to send the packets. In this case, all the packets for the point-to-point communication have to be sent to the border router.

RPL does not allow for a mix of storing and non-storing modes in the same network. It has been shown that this can cause packets to bounce between the two nodes on the path that have different MOP, and never reach the destination [Ko+14].

A simple optimization allows for a node to send a packet directly to the destination, if this is a one-hop neighbor.

2.5.1 Control Packets

In order to construct and maintain the routing topology, RPL defines four types of ICMPv6 [CDG06] control messages:

- `DIO`: DODAG Information Object. This control message contains information about the DODAG, such as the *DODAG Identifier*, the *RPLInstanceID* or the routing metrics used to compute the routes. The messages are broadcasted through the network to construct and maintain the DODAG. The border router is the only node that can start their dissemination. They can also be sent in unicast, to answer a `DIS` message received from one of the neighbors.

- `DIS`: DODAG Information Solicitation. A node may send this message in unicast or in multicast to its neighbors, to proactively solicit configuration information. In response, it will receive a `DIO` packet.

- `DAO`: Destination Advertisement Object. These messages are optional and used only when downward route are needed. They are sent in unicast to selected neighbors (in storing mode) or to the border router (in non-storing mode). `DAO` is the only control message that can be acknowledged by the destination.

- `DAO-ACK`: Destination Advertisement Object Acknowledgment. This packet indicates if the neighbor that received the `DAO` is willing to act as a previous hop for the sender in the downward route. It is sent as a unicast packet in response to a `DAO` message.

2.5.2 Topology Construction

We will present in this section how the DODAG is constructed and which are the routing metrics defined to be used with RPL.

Routing Metrics

The routing metrics that may be used by RPL to construct the DODAG, are [Vas+12]:

- Node State and Attribute (NSA): it provides information on the characteristics of the node, e.g., CPU overload, lack of memory, etc. It can be used to announce which nodes in the network should be avoided;

- node energy: it characterizes the power source of the node (mains powered, battery or scavenger). It can also represent the remaining battery energy;

- hop count: it computes the number of hops on the route, between the source and the destination;

- link throughput: the range of throughput that the link can handle in addition to the currently available throughput;

- link latency: the time interval between the moment a packet is sent and the moment it arrives at the destination;

- link reliability: it can be a discrete value, from 0 to 7, where 0 indicates that the link quality is unknown, and 1 reports the highest link quality. It can also be expressed as the expected number of transmission of a packet necessary for it to be received without error at its destination (ETX [Cou+05], which we detail in the next section);

- link color: semantic constraint used to avoid or attract some specific links. For example, it can signal the links which are encrypted.

Objective Function and Rank

The DODAG construction is based on the *rank* of a node, which depicts its relative distance to the DODAG root. An *objective function* defines how one or more metrics should be used to compute this rank. Also, it specifies the rules that a node has to follow to choose its preferred parent. The IETF Routing Over Low power and Lossy networks (ROLL) Working Group has defined, until now, two objective functions:

- the Objective Function Zero (OF0) [Thurc] is the default objective function and MUST be supported by all the nodes in the network. A node computes its rank by adding a scalar value to the rank of its preferred parent. This scalar value can vary with a ratio from 1 (excellent) to 9 (worst acceptable) to represent the link properties. If OF0 uses the hop count as a routing metric, all the scalars are equal to 1, and shortest routes to the sink are computed.

- the Minimum Rank with Hysteresis Objective Function (MRHOF) [Gna12] also minimizes a routing metric. In addition, it uses a hysteresis to reduce churn in response to small metric changes. Since it can be used only with additive metrics along a route, nodes must support at least one of these routing metrics: hop count, latency, or ETX.

DODAG Construction

The DODAG bootstraps when the border router broadcasts a `DIO` to announce the configuration parameters. When the neighbors of the root receive this packet, they insert the transmitter in the list of possible successors. They choose their preferred parent and compute their own rank with the objective function, and then start broadcasting their own `DIO`. All the nodes in the network follow these steps. The network is configured when all the nodes have chosen their preferred parent, forming this way a DODAG.

Figure 4.8 presents the construction of a DODAG using OF0 with the hop count as a routing metric. The border router starts the dissemination of the `DIO` (Figure 4.8a). Its neighbors choose it as their preferred parent, since it is the closest in number of hops. Then, they start broadcasting their own `DIO` (Figure 4.8b). At the end, all the nodes have a preferred parent (Figure 4.8c).

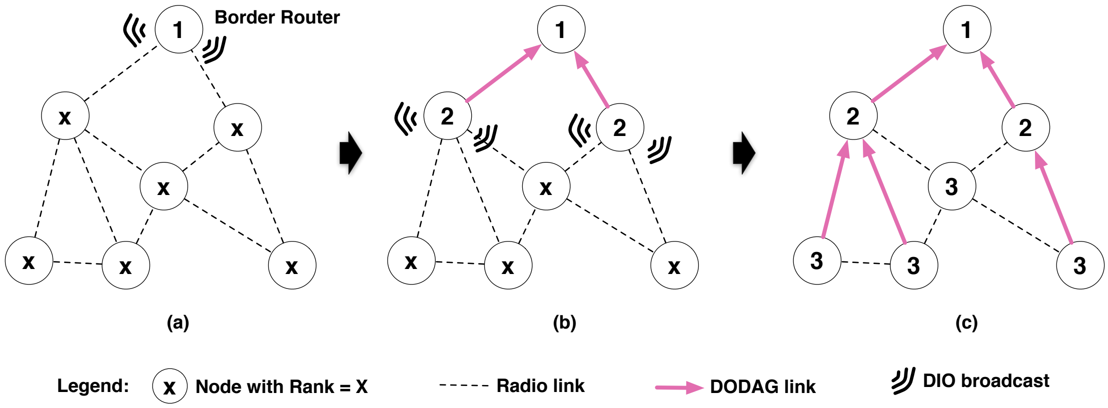


Figure 2.9: DODAG construction using hop count as a routing metric

The Trickle Algorithm

Even after RPL has converged, the protocol keeps on transmitting `DIO` in order to maintain the DODAG up-to-date. The rate at which the `DIO` are being sent is dynamically tuned, by using the Trickle algorithm [Dev+11]: the more stable the topology becomes, the less control messages are being sent, in order to save energy. When an inconsistency is detected Trickle resets its timer and the nodes send `DIO` more frequently in order to quickly propagate the updated DODAG.

2.5.3 Loop Avoidance and Detection

Even though RPL constructs a DODAG, loops may appear because of the variations in the physical connectivity. However, RPL includes rank-based data-path validation mechanisms for detecting loops. We will present in this section how they can be avoided and how they can be fixed (once they have been detected).

Prevention

In order to construct and maintain a loop-free DODAG, the rank of the nodes must strictly monotonically increase from the sink, toward the leaves. Because of this, a node is forbidden to process a `DIO` from a neighbor advertising a rank higher than itself.

RPL also disallows greediness, i.e., a node cannot artificially increase its rank in order to improve some metric or to increase the size of its parent set.

Detection

Loops may occur when `DAO` messages that invalidate a previously announced prefix are lost, or when the `DIO` packets are lost. For example, a node can detach from its preferred parent, and then reattach to a neighbor situated in its prior sub-DODAG.

Such loops can be detected when a node notices an inconsistency between the rank of the node that sent the data packet, and the routing decision for that packet (upward or downward). For example, if a node receives a packet marked as moving downward, from a neighbor with a higher rank than itself, it can conclude that the DODAG is inconsistent.

DODAG Repair

When a loop is detected, a node triggers a local repair. One example of local repair is the route poisoning mechanism. A node detaches from the DODAG, advertises a rank with `INFINITE_RANK` (i.e., it poisons its routes so that the nodes in its sub-DODAG deletes it from their parent set), and then reattaches to the DODAG.

A second DODAG repair mechanism can be initiated only by the DODAG root: the global repair. This is done by incrementing the DODAGVersionNumber in order to initiate a new DODAG Version.

2.5.4 Existing Implementations

Since RPL is the only routing protocol standardized for WSNs, it has been implemented in the main operating systems:

- ContikiOS [Con]: this implementation offers the possibility to have multiple instances, but it does not implement the non-storing mode;
- TinyOS [Tin]: together with the de-facto routing protocol, CTP [Gna+09], TinyOS offers also an implementation of RPL. Some optional features, such as the security mechanisms are not provided;
- RIOT [Rio]: it is a new operating systems, which offers an almost full RPL implementation. The main drawback is that multiple instances are not yet supported;
- OpenWSN [Ope]: being still under development, it does not include yet all the mechanisms of RPL (e.g., the `DIS` messages are not implemented, neither are the security mechanisms).

Implementations of RPL also exist in the main simulators, but not included in the main release: OMNET++ [Var01] *, WSNet [Wsn] † and NS-2 [Ns2; Tri+10].

We present in Table 2.1 a summary of the extensive evaluation of RPL until now. A first thing that we can notice is that although RPL is implemented in all the WSNs operating systems, more than most of the evaluations are done in a simulation environment. Also, few of them use a MAC protocol with duty-cycle, and a single one uses IEEE 802.15.4 with beacon enabled. This is very unfortunate, considering that WSNs must operate on constrained resources.

Even though several link and node routing metrics have been defined for RPL, researchers use mainly ETX and hop count. The reason lies behind the objective functions already defined: MRHOF has been specially designed for ETX, while OF0, together with hop count, need the lowest computational complexity and overhead.

2.5.5 Performance Evaluation

Accetura *et al.* evaluated RPL by simulating networks with 20 and 100 nodes, using the Cooja simulator [Acc+11]. They showed that the protocol induces a lot of overhead, compared to the data traffic. The reason is mostly due to the `DAO` packets, which are used to construct downward routes.

Moreover, Clausen *et al.* highlighted that in storing mode, the nodes closer to the border router will have to store the paths to almost all the destinations in the network [CHP11]. Since the sensors are constrained devices, this might be problematic when the networks are large.

To see how RPL would perform in different home-deployments, Han *et al.* made several experiments where the network was subjected to wireless interference [HG13]. They deployed 25 nodes running on IEEE 802.15.4 channel 12. Then, they added different numbers of Wireless Access Points running on WiFi channel 1, which overlaps with the channel used for the nodes. The results showed that RPL can still reach over 90% of delivery ratio in case of normal interference. Still, it drops drastically (below 10%) when both WiFi and IEEE 802.15.4 interference are present.

*Implementation of RPL for OMNET++: `https://sites.google.com/site/carlesgomez/home/code`

†Implementation of RPL for WSNet: `https://srcnet.u-strasbg.fr/git/oana-rpl-wsnet`

Table 2.1: Summary of RPL Experiments and Simulations*

Ref	**Environment**	**Nodes**	**MAC layer**	**Traffic**	**Routing metric**
[Tri+10]	OMNET++	86	IEEE 802.15.4	6 ppmin & 3 ppmin	ETX
[Wan+10]	NS-2	1000	IEEE 802.11	nodes: 1 ppmin sink: 6 pph	ETX
[Xie+10]	NS-2	1001	IEEE 802.15.4	N.A.	N.A.
[Acc+11]	Cooja	20, 100	IEEE 802.15.4 + LPL	1 ppmin	ETX
[CHP11]	testbed NS2	69 80	IEEE 802.15.4 IEEE 802.11	N.A.	hop count
[HC11a]	NS-2	63-1000	IEEE 802.11 b	sink: 1 pkt to each node	hop count
[HC11b]	NS-2	100 mobile	IEEE 802.11 b	N.A.	hop count
[Ko+11]	TinyOS	51 (TelosB)	IEEE 802.15.4	12 ppmin & 6 ppmin	ETX
[PTD11]	WSNet	256	IEEE 802.15.4 beacon enabled	8 pph	ETX
[DDB12]	ContikiOS	100 TelosB	CSMA	1 ppmin	ETX
[Gad+12]	COOJA	10-100	N.A.	N.A.	ETX hop count
[GK12]	ContikiOS	20 (TelosB)	ContikiMAC	1 ppmin	ETX
[Kam+12]	Cooja	20	ContikiMAC	1 ppmin & 6 ppmin	energy ETX
[Cha+13]	Cooja	4-6	X-MAC	N.A.	custom †
[DLV13]	ContikiOS	135 (TelosB)	ContikiMAC	15 pph	ETX
[HG13]	TinyOS	23	∼ IEEE 802.15.4	1 ppmin	hop count
[HMA13]	ContikiOS	100	sicslomac	80 pph	ETX
[PPR13]	Cooja	17	IEEE 802.15.4	different values between 60 ppmin & 6 ppmin	delay based
[Tia+13]	Cooja	15	CSMA	30 ppmin	geographical information
[Tri12]	OMNET++	15 2442	IEEE 802.15.4	6 ppmin	ETX
[Tri13]	OMNET++	200-2442	IEEE 802.15.4	nodes: 2 packets per day sink: 1 packet every 2h	N.A.
[VTD13]	Cooja	25	ContikiMAC	nodes: 7 ppmin sink: 10 pph (burst)	N.A.
[XL13]	NS-2	256	IEEE 802.11 b	nodes: 120 ppmin sink: 60 ppmin	energy
[Gua+14]	Cooja	9 25	N.A.	different values between 60 ppmin & 1 ppmin	ETX

*N.A. – information not available; ppmin – packets per minute; pph – packets per hour

†$0.5 \times \frac{\text{ETX}}{\text{Maximum\_ETX}} + 0.5 \times \left(1 - \frac{\text{Residual\_Energy}}{\text{Maximum\_Energy}}\right)$

2.5.6 Comparison with Existing Protocols

A comparison with the TinyOS de-facto routing protocol, CTP, is made by Ko *et al.* [Ko+11]. While both protocols obtain similar packet delivery ratio, RPL encounters a slightly higher overhead, mainly because it has more parent changes. They conjectured that this difference in network stability was due to the different routing metrics used by the two protocols. CTP uses Four-Bit [Fon+07], while RPL uses ETX.

Herberg *et al.* made a comparison of the RPL and LOAD routing protocols, when bi-directional traffic flows are present [HC11a]. While they both have the same delivery ratio, RPL manages to offer better end-to-end delay than LOAD. However, RPL induces double the overhead of LOAD, mostly because of the `DAO` packets.

A comparison with the more recent recent version of this protocol, LOADng, was done by Vucinic *et al.* [VTD13]. RPL clearly outperforms LOADng in terms of delay, especially when the nodes are situated further away from the sink. Contrarily to the previous case, the overhead induced by LOADng is higher than the one induced by RPL. This can be explained by the fact that here, the convergecast traffic is predominant, as we can see in Table 2.1.

2.5.7 Convergence and Network Dynamics

In real-life environments, message losses are frequent. Clausen *et al.* demonstrated that, as a consequence, Trickle resets are frequent, making the algorithm converge less well [CHP11].

Tripathi *et al.* [Tri+10] simulated RPL using topology and time-varying link quality data gathered from a real-life outdoor deployment. Their evaluation showed that upon link failure, 90% of the nodes will reestablish connectivity with the DODAG after a maximum of 2 minutes. No local repair mechanism was used, but a global repair was started every 10 minutes. This is very costly in energy, since the DODAG has to be reconstructed every 10 minutes.

They also showed that the Trickle algorithm creates waves in the number of control packets generated in the network: it never self-stabilizes. A more recent survey on RPL [GK12] also brought to attention the problem of oscillating routes when using the objective function based on ETX. This problem was formulated by Gaddour *et al.* as the need for designing new objective functions [Gad+12].

2.5.8 Improvements and Companion Specifications

Wang *et al.* proposed some improvements to the RPL implementation in order to be used for the Advanced Metering Infrastructure (AMI) in Smart Grids [Wan+10]. For example, it uses the traffic generated by the meters to construct downward routing tables, instead of the standard `DAO` messages, reducing thus, the overhead.

Pavkovic *et al.* [PTD11] are the first ones to propose an opportunistic version of RPL. Instead of forwarding the packets just to the preferred parent, a node sends a data packet to the first available parent, without waiting for the preferred parent to be available.

Another opportunistic version of RPL, ORPL, focuses on the point-to-point communication [DLV13]. The authors propose to use anycast transmissions over a low-power listening MAC protocol. Hence, a node does not need to choose a next hop, as the next hop elects itself while receiving. Experiments showed that ORPL increases the latency considerably, while keeping a low energy consumption.

Since the computation of the time at which the `DAO` messages should be sent is not specified by the protocol, Tripathi *et al.* proposed a `DAO` timer mechanism [Tri13]. Each node uses the round-trip time of the `DAO` to determine if there are congestions along the path. Then, in function of this, they will delay or advance the value of the `DAO` timer.

An extension to RPL that improves the point-to-point communication was published as a companion standard. The P2P-RPL uses a reactive technique to construct point-to-point routes [Goy+13]. When a source node needs to discover a path toward another node, it piggy-backs a *route-request* message in its `DIO`. The `DIO` will be disseminated in the network using the

Trickle timer. In consequence, a temporary DODAG will be created, rooted at the source node. This DODAG will cease to exist after a time specified in the Route-Request message.

2.5.9 Current Limits and Drawbacks

RPL has been designed to meet the specifications of different types of applications, including industrial applications. Some of these applications require a minimum latency that can be as low as 100 milliseconds [Pis+09]. Still, this is not easy to achieve in practice with RPL, especially when the nodes are more than 2 nodes away from the sink.

Since its standardization, RPL has been thoroughly evaluated, but without taking into account its energy efficiency. Even though different routing metrics have been defined for RPL, researchers use mainly ETX and hop count (cf. Table 2.1). As severals studies have shown, new routing metrics and objective functions should be defined that will increase the stability of the routing topology and minimize the energy consumption.

We will present next a state of the art of the existing routing metrics in Wireless Sensor Networks, which could be used with RPL.

2.6 Routing Metrics

Researches have proposed multiple routing metrics, trying to answer the different needs of the various WSN applications. We want to present here a state of the art of the existing routing metrics, emphasizing on the energy consumption they require. We categorize them in function of their most prominent component, and we summarize them in Table 2.2.

We can notice that in order to reduce energy consumption, almost all of them use a passive measurement technique, i.e., no control packets are used. Also, the delay constraint is not a primary concern when designing a routing metric. Indeed, the metrics that account for the delay are mostly the ones specially thought for delay-sensitive WSN applications.

In short, a good routing metric should satisfy the following properties:

- capture the variations of the link quality (dynamic);
- maximize the reliability;
- minimize the energy consumption of the nodes.

2.6.1 Physical Layer Metrics

The physical layer offers the less energy consuming metrics. The Received Signal Strength Indicator (RSSI) and the Link Quality Indicator (LQI) are two popular estimators of the radio link quality provided by the radio chipset. All the radios that are IEEE 802.15.4 compatible have to implement the LQI, which can make more precise estimations than RSSI, but it adapts slower to changes in the medium.

The RSSI represents the strength of the received radio frequency signal, and permits only to isolate very good radio links. The gray zone, where the packet reception rate varies considerably, is not differentiable [SL06]. To accommodate the asymmetric nature of links, the RSSI, as well as the LQI, must be measured in both directions.

2.6.2 Link Quality Based Metrics

In order to be energy efficient, the routing protocols have to find the least-expensive paths. One indicator for the energy cost of a path is its reliability. Since packet retransmission is expensive, the routing protocols need good estimators of the link quality.

De Couto *et al.* [Cou+05] have presented a pioneering piece of work to assess the link reliability in a radio environment and has been widely used with RPL [Gad+12]. The Expected Transmission

Table 2.2: Summary of Routing Metrics

			Uses information about			
Name	**Measurement technique**	**Maximizes network lifetime**	**Residual energy**	**Link reliability**	**Retrans-missions**	**Delay**
RSSI, LQI	Passive	No	No	Partially	No	No
ETX [Cou+05]	Active Passive	No	No	Yes	Yes	Indirectly
RNP [Cer+05]	Active	No	No	Yes	Yes	Indirectly
ETT[DPZ04] WCETT[DPZ04]	Active	No	No	Yes	Yes	Yes
Four-Bit [Fon+07]	Passive	No	No	Yes	Yes	No
Residual energy [Kam+12], [XL13]	Passive	No	Yes	No	No	No
Energy + ETX [Cha+13]	Passive	No	Yes	Yes	Yes	Partially
REER [YBo09]	Passive	No	Yes	Partially	No	Partially
Path efficiency [Deh+08]	Passsive	Yes	Yes	Yes	Yes	Yes
REDR [Yoo+10]	Passive	No	Yes	No	No	No
Linear [CT04]	Passive	Yes	Yes	Yes	No	No
RTT [Ady+04]	Active Passive	No	No	No	Yes	Yes
PktPair [Kes91]	Active	No	No	No	Yes	Yes
RPAR [Chi+06]	Passive	No	No	Yes	Yes	Yes

Count (ETX) estimates the number of required transmissions before a correct acknowledged reception. It is computed as:

$$\text{ETX} = \frac{1}{(PDR_{s\to d} \times PDR_{d\to s})} \tag{2.3}$$

where $PDR_{s\to d}$ is the estimated packet delivery ratio from s to d.

ETX is often actively estimated by sending probes to the neighboring nodes, the receiver reporting back the number of received probes. The probes can be sent in unicast or in broadcast, depending on the implementation. However, this method generates a large overhead, since the probes and the data packets must have the same packet size to be accurate. Alternatively, a passive measurement is achievable by piggybacking sequence numbers in data packets. However, this method can estimate the ETX only for radio links already used by the routing protocol. Alternative neighbors cannot be investigated.

Variants have been proposed to deal with other specific constraints, especially in Wireless Mesh Networks. For example, the Expected Transmission Time (ETT) [DPZ04] improves ETX by considering different transmission rates and packet sizes. In order to accommodate multichannel networks, the Weighted Cumulative Expected Transmission Time (WCETT) [DPZ04] has been proposed. Besides considering links which do not use the same channel, it also takes into account the intra-flow interference. Still, both of these metrics are depended on the MAC layer.

The Effective Number of Transmissions (ENT) [KB06] was proposed for the need of higher layer protocols (like TCP) to have a tolerance limit on the number of retransmissions. In other words, ENT minimizes the number of successive link layer retransmissions, to obtain the end-to-end reliability desired by the higher layers.

Cerpa *et al.* proposed to consider the underlying distribution of packet losses [Cer+05]. They have shown that when two radio links have similar qualities, the one with discrete losses should be preferred to the one with consecutive losses. Still, the proposed metric, RNP (the Required Number of Packets), has less good performance results than ETX [Liu+09].

The Four-Bit metric [Fon+07] adopted a cross-layer approach, combining metrics from different layers. From the physical layer, it uses the LQI to get the channel quality during a received packet. From the link layer, it uses the acknowledgement for a transmission, to get the reliability of the link. The last 2 bits are set by the network layer. The *pin* bit does not allow the estimator to eliminate a link that is currently in use. Finally, the estimator can ask the network layer (through the *compare* bit) if the route provided by the sender of the packet is better than the routes already existing in the link table.

2.6.3 Residual Energy Based Metrics

A node should avoid having as next hop a neighbor with a low residual energy, in order to improve the network lifetime. Kamgueu *et al.* [Kam+12] and Xu *et al.* [XL13] have proposed to use RPL with a residual energy metric. However, they do not consider other aspects, such as the radio link quality (and thus, the energy budget for a correct reception). A node might spend a lot of energy retransmitting, if the quality of the link is bad, which overall does not increase the network lifetime. Chang *et al.* decided to combine the residual energy with the ETX into a weighted function [Cha+13]. Even though this metric is simple to implement, it is not directly related to the real lifetime of a node.

REER [YBo09] combines the residual energy, the buffer size (for delay considerations) and the Signal-to-Noise Ratio (SNR) (for link reliability estimation) into a weighted function, to obtain an energy efficient metric. This combination assumes implicitly that the energy consumption depends linearly on these factors, which is not the case. Besides, SNR is complicated to obtain in practice, and it is only loosely correlated with the link reliability [Bac+12].

Another similar approach combines the residual energy, the buffer size, the transmission delay, and the packet reception rate [Deh+08]. Contrary to REER, the information for the link reliability is more accurate, since it is based on the packet reception rate, and not on the SNR. However, the computational complexity is higher (it is not computed just as a weighted function). The metric manages to improve the load-balancing in the network, while reducing the delay.

Yoo *et al.* proposed to use the residual energy depletion rate (REDR) [Yoo+10] to avoid overloaded nodes. REDR is estimated through an exponential moving average of the depletion ratio per second, of the residual energy, during an interval of time. The next hop is chosen as the node advertising the smallest weighted sum between the maximum value of the REDR, and the sum of all the REDRs on the path. Even though this solution considers the residual energy of a node, it does not maximize the lifetime of the network.

Chang *et al.* formulated the routing problem as a linear programming problem where the objective is to maximize the network lifetime [CT04]. They consider the residual energy, the energy spent by a node to send and receive packets, and the transmission rate. This solution could be more effective if the quality of the links was taken into account.

2.6.4 Delay Based Metrics

Some applications may require low delays (e.g., alarms). Several routing metrics have been proposed in this sense. For example, the Per-hop Round Trip Time (RTT) [Ady+04] estimates how much time is required before a packet is correctly received by the destination. It is computed as the difference between the time a packet was generated and the time its acknowledgment was received. This metric indirectly takes into consideration the load of a node and the number of retransmissions due to a bad radio link or interfering nodes.

Since packet length practically impacts the RTT, the Per-hop Packet Pair Delay metric (PktPair) [Kes91] was proposed. PktPair estimates the delay by sending periodically two back-to-back probe packets of different lengths. The receiver computes the difference between the reception time of the two packets and reports this value back to the sender. Considering the amount of control packets that this technique requires, it is not very suitable for WSNs. It greatly increases the energy consumption.

A more effective method to reduce real-time communication delay in WSNs is to dynamically adapt the transmission power of the sensors [Chi+06]. Packets that do not have an urgent deadline are transmitted at a lower power, to reduce energy consumption. When deadlines are tight, a node increases the transmission power to reduce the route length. The computation of the energy estimator is very accurate, accounting for the retransmissions at the link layer. However, even if this solution is energy efficient, it does not maximize the network lifetime.

2.6.5 Summary

Several routing metrics have been proposed in the literature for Wireless Sensor Networks. We presented here a non-exhaustive classification, emphasizing on the energy consumption they require. However, these routing metrics have not been evaluated with RPL. Indeed, until now researchers mainly focused on the hop count and ETX.

Besides the used routing metrics, the success of a routing protocol also depends on the techniques that it uses. For example, RPL, even though it constructs a DODAG, a topology that favors multipath algorithms, it only uses a tree to route. Indeed, one single path is used to route the packets (through the preferred parent), the other ones are maintained just for backup.

Let us present next a state of the art for multipath techniques and their benefits.

2.7 Multipath Solutions

Multipath routing has been widely used in the literature to improve the fault-tolerance (reliability), to balance the load (congestion avoidance), or for QoS improvement [Rad+12]. A node maintains several paths to a destination on which it then splits the traffic load. It can also send all the traffic on a primary route, and use the other ones as backup. The latter is also called alternative path routing.

Considering the improvements that the multipath mechanism can bring to a protocol, the routes may be [TM06]:

- node-disjoint: the paths have no common nodes except the source and the destination;
- link-disjoint: the routes can share some intermediary nodes, but all the links are different;
- non-disjoint (also called braided): there are no constraints on the nodes or links that a route can use. The routes can partially overlap.

2.7.1 Fault-tolerant Routing

Multipath routing reduces the probability that the communication is disrupted in case a node runs out of energy or a link fails. The construction of non-disjoint paths is discouraged, and alternative path routing is preferred.

The Neighbor Disjoint Multipath (NDM) [HSF14] constructs the primary path as the shortest path between the source and the sink. The set of alternative paths are chosen such that the traversed nodes have a small correlation with the nodes on the primary path. The correlation factor is given by the radio range from the nodes on the primary path. For example, if a node is a 1-hop neighbor to a node on the primary path, it has a high correlation. The goal of the algorithm is to minimize the impact of a co-located node or link failure. However, this algorithm is centralized and needs a full knowledge of the network.

The Label-based Multipath Routing (LMR) [HTK04] constructs alternative paths by having the nodes in the network broadcast control packets toward the sink. Every time a node on the primary path has to forward a control packet, it increments its *label*. If the packet received by the sink has a *label* = 0, it means it discovered a node-disjoint path towards the source of that packet.

Since the node-disjoint and link-disjoint paths are usually longer, they are also more energy consuming. Contrarily to the previous approaches, Ganesan *et al.* proposed a braided multipath routing protocol to increase the fault tolerance in WSNs [Gan+01]. Each node on the primary path sends an *alternative reinforcement* message to its next preferred neighboring node, in order to establish alternative paths. Even tough this are just partially disjoint paths, their results show the approach is highly resilient to failure, while being energy efficient.

2.7.2 Energy Efficient Load Balancing

Considering the constraints of the sensors, multipath routing can help prolong the network lifetime by balancing the network traffic in an energy-efficient way.

Chen *et al.* were the first ones to apply energy-balancing routing in WSNs [CN06]. Periodically, the sink selects the path on which to send the data packets, based on the energy level of the different paths it maintains. The authors proposed to compute the optimal paths in a centralized way, by collecting the whole radio topology at the base station. Since the radio topology may change dynamically, and the overhead is important, this approach may perform poorly in practice.

Ming-hao *et al.* adopted the same objective, but they reactively constructed two paths per source [MhRlXd11]. After the first path has been discovered, the source tries to find an alternative node-disjoint path, by searching non interfering nodes. The interference range of a node is assumed to be equal to its radio range. The paths are constructed using a reactive technique, which does not exploit the convergecast traffic pattern of WSNs.

In order to be energy efficient, Yahya *et al.* proposed to consider the radio link quality and the residual energy when computing the paths. They used the REER metric [YBo09] to choose node-disjoint paths. They also computed the number of required paths to successfully deliver a packet, in function of a desired bound of data delivery.

Kacimi *et al.* proposed a load balancing solution where they formalized the lifetime optimization as a nonlinear problem with linear constraints [KDB13]. Even though their method is defined for the convergecast traffic, they assume constant and uniform link quality, which is not the case in real life deployments.

An opportunistic routing specific for WSNs has been presented by Ghadimi *et al.* [Gha+14]. Based on the number of end-to-end duty-cycled wakeups until a packet has reached its destination, each node selects a number of potential forwarders. For each packet transmitted, a single node is selected opportunistically as a unique forwarder. Such algorithms are tightly coupled with a particular MAC protocol.

PWave on the other hand, adopted an analogy with potential fields (similar to a virtual distance) [LSF07]. The protocol constructs multiple routes and balances the load proportionally to the inverse of the cumulative path cost. The authors consider they optimize the energy consumption if the protocol uses the residual energy as the routing metric. However, as we have seen in the previous section, other factors have to be taken into account, e.g., the link quality.

2.7.3 QoS Improvement

Multipath routing improves the QoS support in terms of end-to-end latency and packet delivery ratio. For example, because of the existence of alternative paths when a node/link fails, they reduce the end-to-end delay. The packets are re-routed on other paths and no extra time is spent setting them up. Also, time-sensitive packets could be transmitted on paths minimizing the end-to-end delay, while the other packets on more energy efficient routes.

Felemban *et al.* combined information from the MAC and network layers to create the Multi-Path and Multi-SPEED Routing Protocol (MMSPEED) [FLE06]. The protocol wants to provide QoS support by using geographical routing (the nodes know theirs GPS position), and hence, without taking into consideration the energy constraints of WSNs. MMSPEED assigns each data packet to a *speed* queue, in function of its specified end-to-end deadline. Then, using information from the MAC layer, the time-sensitive packets are prioritized and sent on shorter paths.

Because of its improvement in delay, multipath protocols are also very suited for WSNs multimedia applications. Having this in mind, Li *et al.* use a metric that combines the ETX of a link with its delay, to discover multiple paths with minimum end-to-end delay [Li+10]. Instead of implementing several *speed* queues like MMSPEED, they just add the time-sensitive packets at the beginning of the queue.

2.7.4 Summary

We have shown that multipath techniques can bring several improvements to a routing protocol. While the DODAG structure of RPL creates and maintains multiple paths towards the border router, it only uses a tree to route the data packets. We think that if RPL uses all the parents for routing, and not just for backup purposes, it can improve the performance of the network.

2.8 The Upper Layers

During our work, we focused on the MAC and network layers. Still, let us briefly describe next the upper layers of the standardized stack presented in Figure 2.1c.

2.8.1 The Transport Layer

The transport layer provides end-to-end reliability for IP networks. The best known transport protocols are the Transmission Control Protocol (TCP) [ISI81] and the User Datagram Protocol (UDP) [Pos80].

TCP offers mechanisms to correctly re-arrange out-of-order packets, to eliminate duplicates, and to discard damaged packets. It also provides a flow-control method to minimize the congestion. Together with the retransmission mechanism that it implements, it is able to offer a high reliability, but at the expense of a high overhead induced by the end-to-end packet acknowledgement and retransmissions. Such overhead may not be sustained by the constraint devices used in WSNs.

Unlike TCP, UDP uses a simple transmission model, i.e., it offers a procedure to exchange messages between devices, with a minimum of overhead. This makes it well-fitted for WSNs. However, it does not provide any of the above-mentioned mechanisms. For high reliability, it relies on the link-layer retransmissions. Also, in case of out-of-order packets, their re-arranging is left to the application layer.

2.8.2 The Application Layer

The OSI model defines the application layer as a means for the user to interact with the software application. The IETF Constrained RESTful Environments (CoRE) working group has been intensively working to provide an application protocol that answers to the specific constraints of the WSNs. The Constrained Application Protocol (CoAP) [SHB14], which was recently standardized, is the result of this work.

The main goal of CoAP is to offer a generic web protocol for constrained nodes and networks. It can be seen as a simplified version of HTTP (which is not optimized for low-power communication) that offers the following features [SHB14]:

- low header overhead and parsing complexity;
- reliable unicast and multicast for UDP;
- asynchronous message exchanges;
- URI and content-type support;

- simple proxy and caching capabilities;
- HTTP mapping.

In order to secure the application layer, it also presents a binding to Datagram Transport Layer Security (DTLS) [RM12], which provides communications privacy for datagram protocols.

2.9 Conclusion

In this chapter, we have set up the stage for the rest of this thesis. We made an overview of the protocols used in the current WSN standardized stack, with a focus on the IEEE 802.15.4 MAC protocol and on the network layer. We also presented the state of the art for RPL, the IPv6 routing protocol, and for the routing metrics and multipath algorithms used in WSNs.

RPL has been evaluated in different WSNs scenarios. However, researchers use mainly ETX and hop count to construct the routing topology. These metrics are not energy efficient and do not maximize the network lifetime. Also, as several studies have shown, ETX induces instability in the network. New routing metrics and objective functions should be defined to increase the stability of the routing topology and minimize the energy consumption.

In the next chapters we will present the contributions of this thesis, starting with tackling the broadcast reliability problem in IEEE 802.15.4 networks.

Chapter 3

Broadcast for IEEE 802.15.4

IEEE 802.15.4-2006 was introduced at first in Personal Area Networks (PANs) and now represents one of the major standards for Wireless Sensor Networks (WSN). Even though an amendment of the standard was made in 2012, we will focus here on the 2006 version, since this was the newest version available at the beginning of this thesis.

The medium access may be either asynchronous (beacon-less) or synchronous (with beacons). Without `beacons`, a node cannot sleep as it may receive a frame at any time. Thus, we focus here on the beacon-enabled mode: a node wakes up and transmits a `beacon` to notify its neighbors (i.e., children) that it is ready to receive frames. By appropriately scheduling the active and sleeping parts, IEEE 802.15.4-2006 limits the number of collisions while allowing for energy savings.

Still, it does not provide a specification for one of the fundamental building blocks for the higher layers: a reliable and energy efficient broadcast mechanism. IEEE 802.15.4-2006 makes just two specifications, without explaining *what a node should do* when it must broadcast a packet to all its neighbors:

1. *any frame that is broadcast shall be sent with its Acknowledgment Request subfield set to zero* [802] else, all the acknowledgements would logically collide;
2. a broadcast message *shall be transmitted immediately following the beacon with the CSMA-CA algorithm* [802].

To efficiently deliver both beacon and broadcast packets, we present here a new mechanism for beacon transmission: the Contention Broadcast Only Period (CBOP). We also propose to use broadcast sequence numbers to make the MAC-layer broadcast delivery more reliable.

Contribution

This chapter presents the following contributions:

1. We introduce a new beacon transmission mechanism that limits the impact of a variable `beacon` length, and allows several coordinators per superframe. As we have seen in Chapter 2.3.2, existent solutions waste a lot of bandwidth. Also, this solution allows us to accommodate a variable number of broadcast packets;
2. We implement a reliable MAC-layer broadcast transmissions by introducing a broadcast sequence number;
3. We propose a method to implement a discovery MAC-layer broadcast that transmits a broadcast packet to non cluster-tree neighbors (e.g., non discovered nodes).

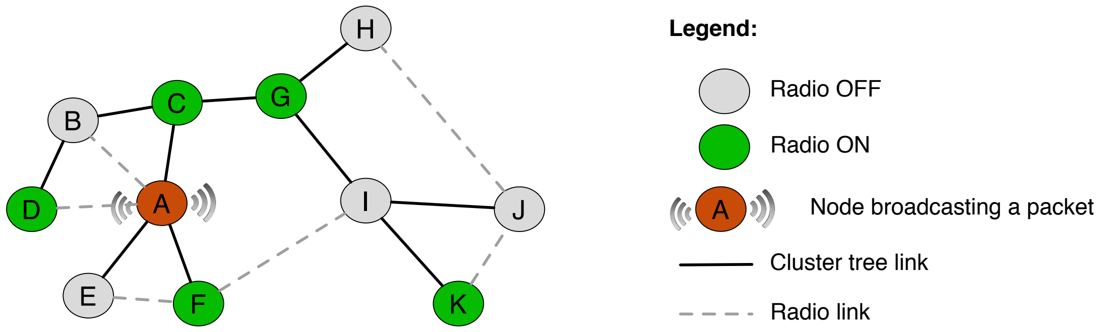


Figure 3.1: Broadcast in a 805.15.4 network

3.1 Problem Statement

In the beacon mode of IEEE 802.15.4-2006, the border router constructs distributively a cluster tree. At the beginning, only the PAN coordinator (i.e., the border router) accepts new associations. Then, iteratively, each newly associated *coordinator* accepts neighbors to associate with the cluster-tree.

In this topology, two nodes that are not associated one with each other, cannot communicate directly. In other words, a node cannot receive a packet broadcasted by a coordinator, if it is not associated with that coordinator. For example, if we take a look at the topology depicted in Figure 3.1, node D cannot receive the broadcasted packet, even though it has its radio on, because it is not associated with the node A.

Also, since the standard does not explain *what a node should do* when it must broadcast a packet, some neighbors will be deaf and will not receive the packet. If we go back to the example in Figure 3.1, we can notice that the child E will miss the broadcast packet, since it has its radio off. All the children must consequently stay awake during the whole active part of the superframe of its parent, in case they receive a broadcast packet.

We consider that the broadcast in IEEE 802.15.4-2006 faces the following problems:

- a MAC-layer broadcast packet may not be received because radio links are practically unreliable. Since some protocols (e.g., RPL [Win+12]) depend on quite-reliable broadcast, this would cause convergence problems;

- since IEEE 802.15.4 exploits a tree, a network-wide broadcast (flooding) requires that each node forwards a broadcast packet. A single transmission failure may severely affect the global reliability, and amplify the well-known unreliability problem in radio networks [Kuh+10];

- a node may turn-off its radio to save energy (e.g., during its backoff or during its active period). However, a coordinator may transmit a broadcast packet at any time;

- no discovery method is proposed. Higher layers protocols use broadcast packets to discover the neighbors of a node i.e., the nodes situated in its transmission range. The discovery method is needed both immediately after the deployment, and when triggered by an event (e.g., a local repair mechanism).

To the best of our knowledge the MAC-layer broadcast problem has not yet been studied in IEEE 802.15.4-like networks. The focus was uniquely given to network-wide broadcast (i.e., network flooding).

3.2 Beacon Collision Avoidance

In the beacon enabled mode, the active periods must be carefully scheduled to avoid collisions among both `beacons` and data packets. If two or more coordinators do not interfere with each other, or if at most one of them has children, they can be scheduled to share the same superframe.

However, reserving a whole superframe slot for one coordinator without children is clearly sub-optimal.

In the Beacon Only Period (BOP), at the beginning of each superframe, several slots are dedicated to `beacons`. Coordinators sharing the same superframe must choose a different BOP slot. A coordinator chooses its BOP slot according to the uniform distribution. Let n_{slots} be the number of BOP slots, and n be the number of coordinators. Let $P[coll]$ represent the probability that at least one `beacon` collision happens. So we firstly compute the probability when no `beacon` collides. We consider that the coordinators are ranked (i.e., by their id). Let us assume that the first coordinator has chosen a given random timeslot. No `beacon` collides if each k^{th} coordinator ($k \geq 2$) chooses a slot different from all other first $(k-1)$ coordinators. In other words, it has the choice among $n_{slots} - (k-1)$ slots. Consequently:

$$P[coll] = 1 - \prod_{k=2}^{n} \frac{n_{slots} - (k-1)}{n_{slots}} = 1 - \prod_{k=0}^{n-2} \frac{n_{slots} - (k+1)}{n_{slots}} \tag{3.1}$$

Thus, n_{slots} has to be chosen large enough to make this probability small.

Figure 3.2 illustrates this limit of the BOP method: the collision probability quickly becomes large when several coordinators compete for a BOP slot. Practically, we must maintain quite a large number of BOP slots, although a BOP slot consumes bandwidth: in a BOP slot must fit a whole `beacon`. A `beacon` may be long since it contains a list of pending frames (short and long addresses). Typically, a BOP slot duration is around $4ms$. If the SO value is small (i.e., the value that defines the length of the active part of the superframe), the Beacon Only Period consumes most of the active part of the superframe. If we neglect the clock drifts, 4 BOP slots with SO = 1 last 36% of the active part: there is not much space for application data transmission.

Since the duty-cycle is equal to $2^{-(BO-SO)}$, we should maintain a small SO to reduce the end-to-end delay, while keeping a small duty-cycle. Unfortunately, this will also reduce the space for data transmission.

We would like to draw attention to the fact that a Beacon Only Period is very important to make the IEEE 802.15.4 network scalable. A conflict-free scheduling for superframes may even be impossible to obtain in large density cases or when the number of superframe slots is limited. However, reserving a whole superframe slot for one coordinator without children is clearly sub-optimal.

3.2.1 Our Solution: Contention Broadcast Only Period (CBOP)

We propose to replace the TDMA solution for the Beacon Only Period with a deterministic contention, which uses different Inter-Beacon Space (IBS) values. While the first one fixes a priori the number of slots with a predefined length, we would rather control the Inter-Beacon-Space to handle several coordinators in a single superframe.

We adopt here an approach inspired from [LW08] where each node chooses a static mini-slot to transmit its frames. While the original approach focused more on single hop topologies, we propose to adopt this approach for multihop IEEE 802.15.4 networks, avoiding collisions among `beacons`.

A coordinator chooses an Inter-Beacon Space (IBS) value, constant for all its `beacons`:

- it randomly chooses one `IBS_value` in the range $[0..b_{max}-1]$,where b_{max} is the maximum backoff period;
- when the superframe begins, a coordinator has to wait for `IBS_value`$*$`aUnitBackoffPeriod` idle time before transmitting its `beacon`.

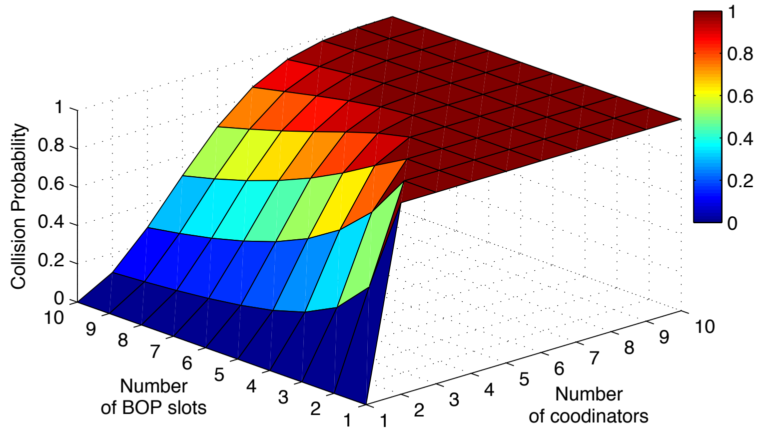


Figure 3.2: Collision Probability during the Beacon Only Period

Two `beacons` collide only if both coordinators have chosen the same `IBS_value`. When this occurs, CBOP adopts the same algorithm to assign collision-free values as the BOP solution.

We can notice that the BOP slot duration is much longer than the IBS value ($4ms >> 0.3ms$). Thus, CBOP will save on average more bandwidth than the BOP strategy.

The Contention Access Period (CAP) starts b_{max} ∗ `aUnitBackoffPeriod` after the transmission of the last `beacon`. When a child senses the medium idle during the maximum `IBS_value`, it can safely consider that the `beacon` period is over.

3.2.2 Synchronization Requirements

Clock drifts have a significant impact on the performance of the BOP approach. In the classical version, a BOP slot must contain the `beacon` and a `guard_time`. The guard-time is obtained via the maximum clock-drift bound and the inter-beacon period [Bac+10a].

In CBOP, we propose the following approach:

1. when a coordinator wakes-up at the beginning of its superframe, it must wait (`IBS_value`+ 1) × `guard_time` before transmitting its own `beacon`. This way, all the coordinators will defer their transmission in order to synchronize, while keeping the relation between their corresponding backoff values;

2. after the reception of a `beacon` from another coordinator, a node has to defer its `beacon` transmission, since the medium could be busy. Still, it will only wait for `IBS_value` ∗ `aUnitBackoffPeriod`. No need to wait for `guard_time`, the synchronization has been done at the moment the coordinator woke up.

For the synchronization, in the worst case CBOP wastes `guard_time`∗b_{max} when a coordinator is alone and has chosen the maximum BOP value. Since the classical BOP version wastes exactly `guard_time` ∗ n_{slots} in any case, we reduce on average the overhead due to synchronization.

3.2.3 Discussion on the Global Bandwidth Wastage

When we have the maximum number of coordinators ($= n_{slots}$) per superframe, CBOP will face the worst case. Lets consider that a `beacon` transmission lasts at most 4ms, and on average 1ms:

- BOP uses $n_{slots} \times (4ms + guard\_time)$ (each BOP slot must have a fixed duration to contain the longest `beacon`);

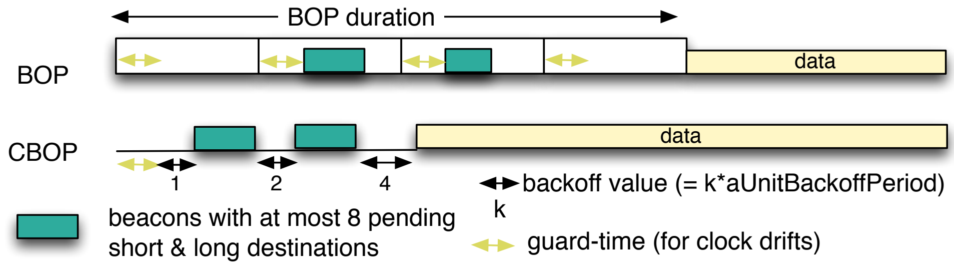


Figure 3.3: Overhead for BOP and CBOP

- CBOP uses:
  1. the guard time to deal with the clock drift, but only once, for the coordinator with $IBS\_value = 0$;
  2. $n_{slots} * 1ms$ for the transmission (one beacon per coordinator);
  3. $\left(\sum_{k=0}^{n_{slots}-1} k\right)$ `aUnitBackoffPeriod` for the backoff of each coordinator.

  Finally, the time dedicated to `beacons` is:

$$guard\_time + 1ms \times n_{slots} + \frac{(n_{slots}-1)(n_{slots}-2)}{2} * \texttt{aUnitBackoffPeriod} \quad (3.2)$$

Since the guard-time is actually longer than the `aUnitBackoffPeriod`, and the number of BOP slots is actually limited, CBOP performs better even in the worst case.

Besides, the CBOP method is more flexible since it can deal efficiently with `beacons` with a variable size (the number of pending addresses included in the `beacons` has a signifiant impact on the `beacon` size).

Let us consider an usual case where two coordinators share the same superframe (Figure 3.3). They both have 2 pending short destinations to include in their `beacons`. With 4 BOP slots, the BOP strategy must reserve at least $4 * 4ms = 16ms$ for the BOP duration: a `beacon` with the maximum packet length must fit in each BOP slot. With CBOP, we must have two `beacons` ($\approx 2ms$ on average) and 3 IBS values ($< 2ms$). Clearly, more bandwidth is wasted for `beacons` in the BOP strategy, compared to our CBOP proposal.

3.3 Reliable and Energy-Efficient Broadcast

As previously highlighted, IEEE 802.15.4 does not exactly specify how to deliver broadcast packets to all the radio neighbors. Bachir *et al.* make a distinction between neighboring broadcast (the packet has to be delivered to all the neighbors) and discovering broadcast (the packet aims at discovering new neighbors) [Bac+10a].

In multihop networks, we must implement both types of MAC-layer broadcast. Indeed, we may require discovering broadcast (e.g., for the cluster-tree reconfiguration) and neighboring broadcast (e.g., for control traffic generated by the routing protocol).

3.3.1 Duplicated Broadcast

The easiest way to implement a reliable MAC-layer broadcast consists of two parts: duplicate the packet and send one copy per child in unicast [Guo+09]. This approach presents two limitations:

- duplicates consume energy: a broadcast transmission is sometimes sufficient to cover several neighbors;
- this method does not work for *discovery*: a coordinator cannot enqueue unicast packets for unknown destinations.

We propose to use our Contention Broadcast Only Period to efficiently disseminate MAC-layer broadcast packets, while not suffering from these limitations.

3.3.2 Broadcast Sequence Number

We will designate the nodes which track the `beacons` of a neighboring coordinator as *followers*. A *follower* may track broadcast packets from a coordinator (e.g., `hellos`) while not being associated with this coordinator. This means that non cluster-tree neighbors can also track and receive the broadcasted packets. Such *follower* MUST listen to the `beacons` of a neighboring coordinator at most every `macTransactionPersistenceTime`.

After having transmitted its `beacon`, the coordinator also sends the broadcast packets buffered since its last `beacon`. This transmission is safe since another coordinator must sense an idle medium before transmitting its own `beacons`. Besides, we are no longer blocked by the fixed duration of a BOP slot, which forbids to send a variable number of `beacons` and broadcast packets.

We can notice that a broadcast packet is transmitted only once, following the `beacon` after the packet has been generated/received. It is NOT transmitted after each `beacon`.

Since a neighbor may miss the broadcasted packet (it is sleeping or the packet was corrupted because of a lossy link), we must also implement a mechanism to guarantee the reliability. Consequently, a coordinator also maintains a Broadcast Sequence Number (BSN) in the `beacon`: each time the coordinator has to send a broadcast to its neighbors, it increments the BSN value and enqueues the corresponding broadcast packet.

Broadcast reliability is achieved in the following way:

1. a coordinator piggybacks in its `beacons` its current BSN value;
2. the node compares the BSN included in the `beacon` and the BSN saved in its neighborhood table. If values differ, it generates a `Broadcast-Request` with the last received BSN (i.e., the *requested* BSN). It sends the packet according to the slotted CSMA/CA algorithm;
3. the coordinator acknowledges the `Broadcast-Request`. Then, it sends back-to-back the enqueued broadcast packets with a sequence number superior or equal to the requested BSN;
4. to exploit the broadcast nature of radio transmissions, the followers that also have a different BSN (i.e., they are missing some of the broadcast packets requested by the other node), will keep their radio on. This way, they can also receive the broadcasted packets re-sent by the coordinator.

Although the broadcast transmissions are not acknowledged, we guarantee the reliability: the BSN value for one follower is only incremented when it receives the corresponding packet. Thus, `Broadcast-Request` will keep on being generated until the packet is correctly received. Besides, if the packet was dropped in the meantime by the transmitter, an empty data packet will be replied.

3.3.3 Fairness

If several coordinators share the same active part, unfairness may appear. Indeed, the coordinator with the smallest IBS may capture the medium for all its broadcast transmissions.

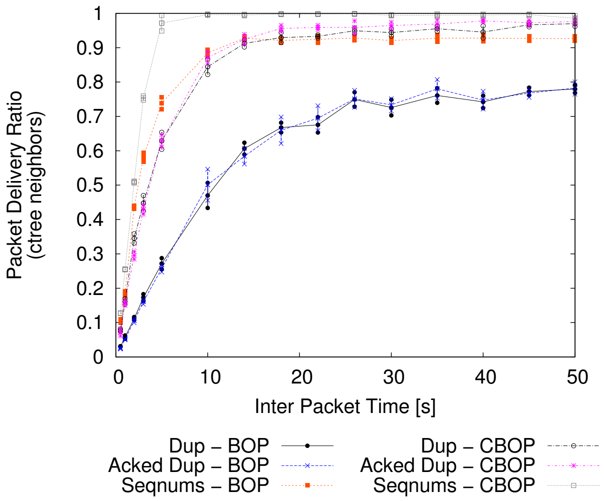


Figure 3.4: PDR with an increasing traffic for the different algorithms

To avoid this scenario, a coordinator computes a fair use of the bandwidth: `Superframe Duration` divided by the maximum number of coordinators sharing an active part (b_{max}). A coordinator cannot transmit broadcast packets for a duration longer than $\frac{SD}{b_{max}}$.

We adopt here a pessimistic approach, considering the maximum number of contending coordinators. We could implement an adaptive approach, where each coordinator counts the number of contending coordinators in its active part. This parameter would be updated at the end of each active part.

3.4 Performance Evaluation

We have used WSNet, an event-driven simulator for large scale wireless sensor networks [‡] to implement the beacon-enabled mode of IEEE 802.15.4. The simulator has been thoroughly evaluated [HCG08]. Each coordinator greedily and distributively selects a superframe slot to limit collisions [PTD11]. The cluster-tree is distributively constructed, coordinators blacklisting `beacons` with a too small RSSI to avoid choosing bad links in the cluster-tree.

We have considered random circular topologies, where the PAN coordinator is located at the center of the simulated area and the other sensors are placed randomly on a disk (on average, a node has 9 neighbors). We consider only Full-Function-Devices (FFD) i.e., any node joining the cluster-tree acts as coordinator. By default, the network comprises 50 nodes. We ran 10 simulations for each set of parameters and inserted the 95% confidence interval in the graphs.

At the PHY layer, we used the path-loss shadowing model, calibrated with the scenario FB6 (indoor real deployment) presented in [CT11] (shadowing, path loss= 1.97, standard deviation= 2.0, $Pr(2m) = -61.4dBm$). We used $BO = 8$, $SO = 1$ (duty-cycle $\simeq 1\%$) and 4 BOP slots.

We compared the following solutions:

- `Dup`: a broadcast packet is duplicated into several unicast packets and sent to the parents and children;
- `Acked Dup`: the broadcast packets are duplicated and each unicast copy must be acknowledged by the destination;
- `Seqnums`: we implemented the sequence number piggybacked on `beacons` together with the `Broadcast-Requests`.

[‡] `http://wsnet.gforge.inria.fr`

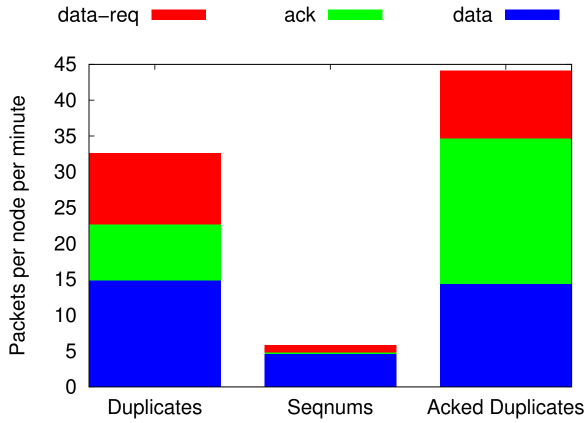


Figure 3.5: Overhead — 1 broadcast packet every 20s, CBOP algorithm

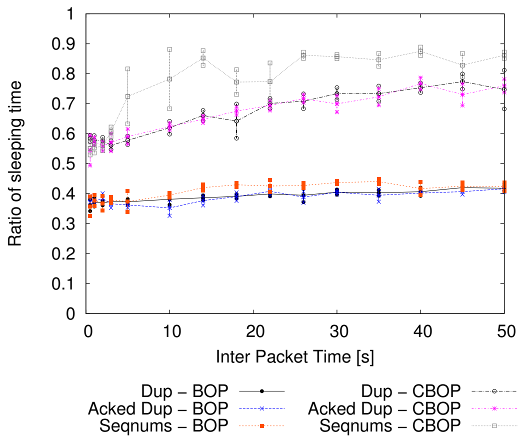


Figure 3.6: Energy — duty-cycle ratio

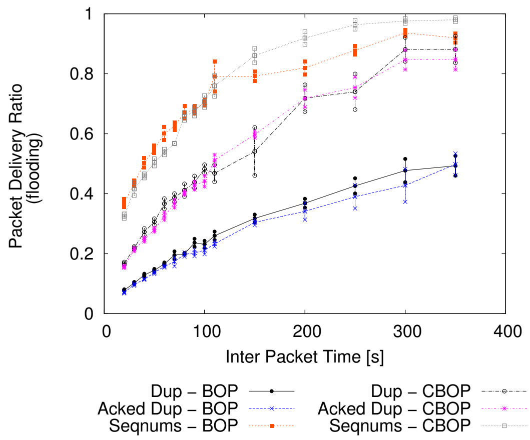


Figure 3.7: Coverage for a flooding

We also compared the original Beacon-Only Period Solution (BOP) — TDMA solution in which one slot is dedicated to each `beacon`— and our Contention Broadcast Only Period mechanism (CBOP).

We first measured in Figure 3.4 the packet delivery ratio for packets broadcasted to the cluster-tree neighbors (parents and children). We can observe that our Contention Broadcast Only Period efficiently disseminates broadcasts while minimizing the bandwidth dedicated to the CBOP (waste of bandwidth due to IBS is limited). On the contrary, the BOP solution creates many collisions among `beacons`, explaining the lower packet delivery ratio. We can also notice that our broadcast solution based on sequence numbers is more efficient compared to duplicating broadcast packets. Moreover, our CBOP mechanism is more robust to larger traffic. With CBOP, we may operate at a lower duty-cycle, increasing energy savings.

We also measured the overhead (Figure 3.5). The seqnum solution efficiently reduces control traffic: broadcast transmissions during the CBOP are often sufficient to *cover* all the neighbors. Some additional `Broadcast-Requests` are seldom required for unreliable links.

We evaluated the impact of the CBOP algorithm on the energy consumption. In particular, Figure 3.6 reports the average sleeping time for each solution. With BOP, a node has to wait for the whole BOP duration for all the superframe it follows. Thus, a node sleeps just around 40% of the time. On the contrary, CBOP limits idle listening by reducing the period dedicated to broadcast and `beacons`, making a node sleep longer (around 80% of the time).

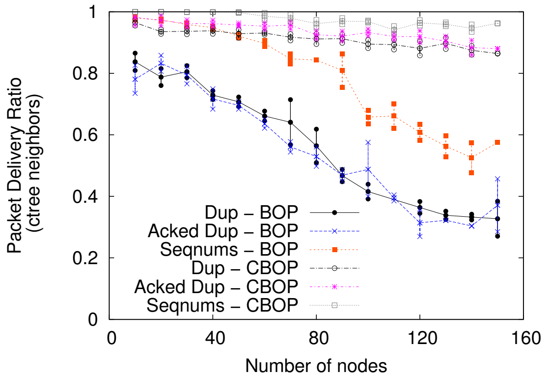


Figure 3.8: Scalability – 1 packet every 7 seconds

Next, we evaluated the impact of the broadcast algorithm on the flooding reliability: each node which receives a broadcast packet has to forward it (Figure 3.7). By exploiting efficiently the redundancy of the flooding structure, CBOP with sequence numbers achieves the best reliability. We can notice that with both methods, creating duplicated packets increases the number of collisions, negatively impacting the reliability.

Finally, we evaluated the scalability of these solutions (Figure 3.8). BOP is not scalable: more nodes mean more collisions among `beacons` and among data packets. Thus, the broadcast reliability quickly decreases when we increase the number of nodes. While CBOP achieves an almost perfect delivery for small networks, duplicating packets often creates more collisions: several children may simultaneously send an IEEE 802.15.4 `data-request` command to retrieve packets buffered at a coordinator. This well-know phenomenon in IEEE 802.15.4 impacts the packet delivery ratio when duplicating broadcast packets.

3.5 Perspectives

The 2012 amendment of the IEEE 802.15.4 MAC standard focuses on multi-channel solutions. Among the several extensions, the Timeslotted Channel Hopping (TSCH) mode is a promising solution.

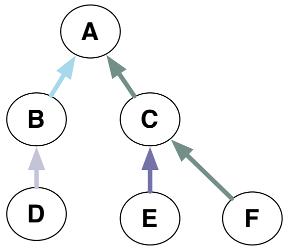
In the TSCH mode, the active period is reduced to the smallest duration possible, by replacing the concept of superframe with slotframe, and by eliminating the `beacons`. As we can see in Figure 3.9b, a slotframe is divided into several slots of equal sizes. A slot is long enough to fit a packet of maximum length (127 bytes) and its acknowledgment. A schedule (centralized or distributed) is used to tell each node what to do in a slot: sleep, receive, or transmit a packet. Moreover, it also specifies to which neighbor it can communicate and on what channel offset (Figure 3.9b).

A *dedicated link* is assigned to a single radio link while a *shared link* may be used by several receivers (without acknowledgment) and/or several transmitters. CSMA/CA or ALOHA is required to solve conflicts between interfering transmitters in a shared link.

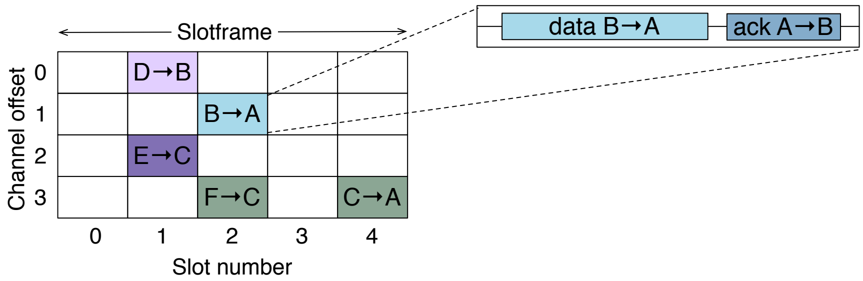
Broadcast in these conditions faces the same problems as in the beacon mode. On one hand, the unreliability caused by the fact that some of the receivers may not receive the broadcast packets. On the other hand, the inefficiency caused by individually acknowledging each broadcast packet by each receiver.

A more efficient broadcast mechanism that ensures the reliability is needed. We may adapt our broadcast solution to the TSCH mode to be used for shared links with several receivers:

- a transmitter piggybacks in e.g., Enhanced Beacons its current broadcast sequence number (BSN);



(a) Cluster tree topology



(b) TSCH scheduling and slot structure

Figure 3.9: IEEE 802.15.4 TSCH

- the broadcast packets are sent during the advertising slots;
- a neighbor may ask the missing broadcast packets to the source during a different timeslot;
- the source finally delivers the required broadcast packets in its next timeslot dedicated to broadcast.

Still, a more thorough study of this method could be made, in an effort to further minimize the overhead and the energy consumption.

3.6 Conclusion

We proposed to modify the IEEE 802.15.4 superframe structure by introducing a Contention Broadcast Only Period: each competing coordinator chooses distributively a fixed Inter-Beacon-Space to send its `beacons` and broadcast packets in its superframe. By removing the BOP slot, we reduce the bandwidth wasted by `beacons`: we can safely reduce the duty-cycle while maintaining the same capacity. Besides, we also proposed to use broadcast sequence numbers to guarantee a certain reliability in lossy networks. Simulations with a realistic shadowing PHY model prove our solution efficiently disseminates broadcast packets while limiting the overhead.

Since we validated our proposal only through simulations, we should also explore the impact of real-testbed deployments on our broadcast strategy. Even though we used a realistic physical layer, the experiments can offer results closer to reality, due to phenomenons that are more difficult to implement in simulation. For example, the impact of clock drifts on our CBOP mechanism is one of the things that we could study.

We still have to investigate what would be the optimal cluster-tree, to efficiently implement both unicast and broadcast transmissions. We must also study how self-pruning techniques may be incorporated to this mechanism for reliable flooding.

Now that we have a reliable broadcast mechanism at the MAC layer, we can focus on the network layer. We start the next chapter by providing an exhaustive evaluation of RPL, under different routing metrics.

Chapter 4

Stability and Efficiency with RPL

RPL is based on the construction of a Destination-Oriented Directed Acyclic Graph (DODAG). Depending on the routing metric that it uses, the network will perform differently in terms of end-to-end packet delivery, delay, energy consumption, and topology dynamics. Indeed, each metric is different in terms of environmental information that it uses, computational complexity, and induced overhead.

While several routing metrics have been proposed in the literature for WSNs, they were not systematically evaluated with RPL. As we have seen in Section 2.5, researchers mainly use ETX and hop count for the construction of the DODAG (cf. Table 2.1). Also, to the best of our knowledge, there is no study comparing the performance of RPL under different routing metrics, in the same setup.

We offer here a thorough evaluation of the behavior of RPL when using different routing metrics for the construction of the DODAG. If RPL does not work in these scenarios, it will a fortiori behave poorly experimentally, facing at least the same problems.

Contribution

This chapter presents the following contributions:

1. We evaluate by simulations the performance of RPL and the stability of the DODAG structure with a realistic physical layer;
2. We present tools and metrics to study the stability of a routing protocol for WSNs, investigating in particular the routing dynamics of RPL;
3. We demonstrate that no existing routing metric succeeds to guarantee both the stability and the efficiency of RPL. Thus, efficient routing metrics still have to be proposed for RPL.

4.1 Problem Statement

RPL is a distance vector protocol designed for networks containing up to thousands of constrained devices. By using different routing metrics, RPL constructs a Destination-Oriented Directed Acyclic Graph (DODAG) rooted at the border router, gateway to the Internet. Hence, choosing these metrics is of uttermost importance for ensuring the performance of the network.

When constructing a DODAG, the most important thing for a node is to wisely choose its parent in order to meet the delay and reliability constraints. Finding the right parent means having a good link quality that will guarantee low packet loss even when the network is highly unstable or the traffic is very high. However, estimating the link quality in a reliable manner and finding accurate routing metrics are still open problems.

A good metric should not induce too much overhead or instability in the routing topology. It should quickly react to changes, but only if these changes are durable. Also, we have to take into account implementation considerations and the tuning effort required.

To the best of our knowledge, there is no study comparing the performance of RPL under different routing metrics, in the same setup. Moreover, most of the implementations use ETX and hop count for the construction of the DODAG (cf. Table 2.1).

4.2 Implementing Routing Metrics for RPL

In Chapter 2.6, we have categorized the routing metrics in function of their most prominent component. To study the behavior of RPL under various routing metrics, we chose to implement one metric of each category that reflects the quality of the radio link. Firstly, we selected the two most widely used metrics: hop count and ETX. Secondly, we chose LQI since it depicts well the radio link quality measured by the radio chipset. However, RSSI would have lead to the same final results.

Because cross-layer metrics highly rely on different metrics, their behavior is more complicated to assess. Moreover, they depend on the MAC layer. Thus, we decided to rather focus on individual routing metrics. Our implementation may be adapted to any MAC layer that implements acknowledgements for data packets.

4.2.1 The Expected Transmission Count (ETX)

We recall that ETX estimates the number of transmissions required before the reception of a correct acknowledgement [Cou+05]. It is computed as:

$$\text{ETX} = \frac{1}{(PDR_{s \to d} \times PDR_{d \to s})} \tag{4.1}$$

where $PDR_{s \to d}$ is the estimated packet delivery ratio from s to d.

In order to save energy, we adopted a passive measurement technique: we estimate the *Packet Delivery Ratio (PDR)* of a link by using the existing data traffic, without sending probes. This technique tends to over-estimate the packet losses in the presence of asymmetric links. Indeed, a data packet may have arrived while the acknowledgement was lost. However, this event has been proved to be experimentally negligible [SAZ10].

We only considered acknowledged packets to estimate the *bidirectional* packet delivery ratio. Every time an acknowledgement is received, a node updates the ETX of that link with the corresponding new PDR. The PDR for a link is computed as the ratio between the number of acknowledgements received, and the number of data packets sent on that link, including retransmissions. If all the retransmissions of a packet have failed, the packet is dropped and the ETX is downgraded accordingly. Since counting the acknowledged packets implicitly combines the PDR in both directions, just one PDR measurement is sufficient to compute the ETX value.

However, such passive technique implies that we can estimate the quality of a radio link only when it is selected by RPL (either as parent or as child). In order to also evaluate the links

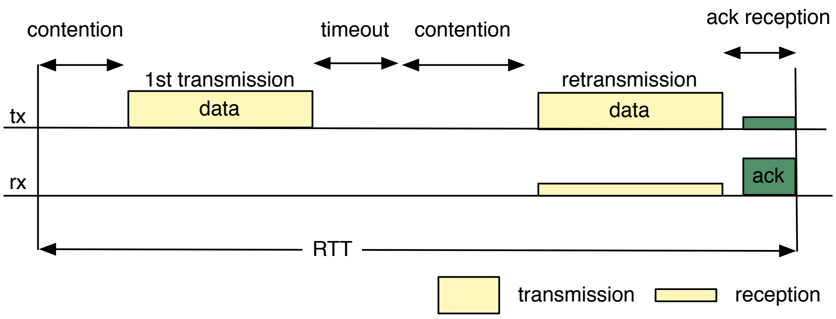


Figure 4.1: Round Trip Time (RTT) estimation

that are not used by the data traffic, we compute an estimation of the ETX by monitoring the received `DIO`. We modified the `DIO` packet in order to contain a *SequenceNumber*, so that the receiver is able to estimate the `DIO` loss ratio. Every time a `DIO` message is received, the node computes the difference between the last received *SequenceNumber* and the new one. It then updates the ETX accordingly.

In other words, a radio link dropping most of the `DIO` will probably never be chosen: we do not waste energy to estimate more accurately its quality. On the contrary, a node may select as parent a neighbor from which it receives most of the `DIO`. Then, it will use the data traffic to refine the link quality metric (an asymmetrical link will surely be removed later).

4.2.2 Delay Based Metrics

When considering delay based routing metrics, RTT captures the latency in the network, while considering the quality of the path. However, we decided to not implement it, since it is closely related to the ETX metric and it would give the same results.

The contention time mainly depends on the number of retransmissions. Figure 4.1 depicts the algorithm followed by a node for the transmission of a packet. If the sender does not receive an acknowledgment before the expiration of a timer, it will try to resend the packet. First, it doubles the value of its backoff (i.e., time to wait before checking if the channel is busy). Then, it sends the packet when it senses that the channel is idle. The retransmissions stop when an acknowledgment is received, or when the maximum number of retransmission is reached.

We can conclude that the RTT may be estimated as follows:

$$\text{RTT} = \sum_{k \in 1}^{N} (T_{contention}(k) + T_{data}) + (N-1) \times T_{timeout} + T_{ack} \tag{4.2}$$

where N is the average number of retransmissions for a packet, T_x the time needed to send a data packet or an ack, and $T_{contention}(k)$ the contention time for the k^{th} retransmission.

Since N = ETX, the ETX and the RTT metrics are consequently very close, and would exhibit the same behavior in single rate networks. Besides, the MAC layer must report to the network layer $T_{contention}(k)$, which is difficult to achieve in practice.

4.2.3 The Link Quality Indicator (LQI)

The LQI is provided by all the radio chipsets that are IEEE 802.15.4 compliant. The standard specifies that the values of the LQI should be uniformly distributed in the range of 0 through 255. However, its exact computation is left to the manufacturer choice.

In the implementation of the IEEE 802.15.4 that we used, the LQI was computed as an estimation of the signal-to-noise ratio for every received packet, be it a data or a control one.

Table 4.1: Simulation parameters

Parameter	Value
Simulation duration	3600 s
Number of nodes	100
Simulated area	600m x 600m
Traffic type, rate	CBR, each node sends 5 pkt/min
Data packet size	127 bytes (incl. MAC headers)
RPL	MinHopRankIncrease = 256
Trickle	$Imin = 128ms$, $Imax = 16$, $k = 10$
Statistical estimator	$\lambda = 0.9$, blacklist-threshold: link quality $\geq 10\%$
MAC layer	IEEE 802.15.4 beacon less

This information is sent then by the MAC layer to RPL. By using this metric, we implicitly consider that the radio link quality is symmetrical.

4.2.4 Statistical Estimator

All these metrics denote instantaneous values. Since they are stochastic by nature, we must smoothen their values in order to limit their variation. We decided to use an Exponential Weighted Moving Average (EWMA) estimator:

$$Metric(t+1) = \lambda Metric(t) + (1-\lambda) measure \tag{4.3}$$

where $Metric(t)$ represents the estimated metric at time t, $measure$ the new measurement of this metric, and λ relates to the *memory* of the metric. A new radio link initially has a metric equal to 1.0. We fixed the value of λ in the performance evaluation section, and showed that it only slightly impacts the performance of the protocol.

We also used a blacklisting policy in order to take into consideration only links above a certain threshold (cf. Table 4.1). The nodes that are blacklisted are removed from the list of possible successors. It was shown by Liu *et al.* in [Liu+09] that it has a positive impact on routing.

4.3 Simulation Setup

We adapted the RPL implementation of Contiki [Con] to the WSNet simulator, an efficient event-driven simulator dedicated to Wireless Sensor Networks, which has been extensively evaluated [HCG08]. By means of simulations, we can analyze more easily the measurements, because of reproducibility, isolation, and control on the parameters. The results are averaged over 20 simulations with different random topologies. We consider Constant Bit Rate (CBR) convergecast flows because this is the most frequent scenario in WSNs.

In order to focus on the properties of the routing protocol, we decided to use at the MAC layer the IEEE 802.15.4 protocol in the beacon-less mode (i.e., the radio is always on). In this way, the behavior of the routing protocol will not be influenced by the duty-cycle. A preamble technique may be used together to implement a low-duty cycle MAC.

At the PHY layer, we used the path-loss shadowing model, calibrated with the scenario FB6 (indoor real deployment) presented in [CT11]: shadowing, path loss = 1.97, standard deviation = 2.0, $Pr(2m) = -61.4dBm$, when the environment is static but accurately modeled.

We configured RPL as illustrated in Table 4.1. $Imin$ (the initial interval between two `DIO` emissions) was chosen to limit the overhead during the initialization phase and every time the Trickle algorithm is reset. $Imax$ (the number of times the interval can be doubled) uses its default value (conform with the chosen $Imin$). At boot-up, the nodes stay silent, that is, they wait for `DIO` from other nodes, without sending `DIS` messages. As the simulated traffic is convergecast (i.e.,

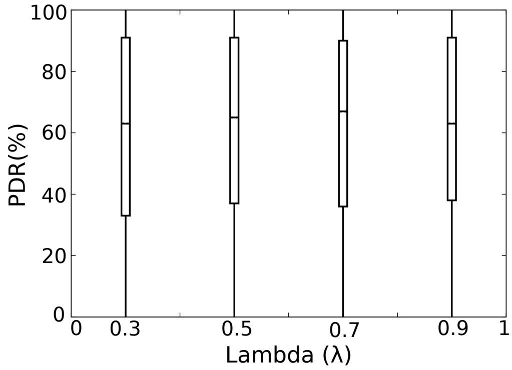


Figure 4.2: λ versus the packet delivery ratio, with ETX

multipoint-to-point), `DAO` messages are deactivated. The objective function used if OF0 [Thurc] for hop count and LQI, and MRHOF [Gna12] for ETX.

Figure 4.2 illustrates the impact of the λ parameter on the reliability of the network. We can notice that the packet delivery ratio is only very slightly correlated with the λ value. In order to limit the impact on the network dynamics, we decided to choose $\lambda = 0.9$. This way, we manage to avoid variations in the metric by not overreacting to transient changes.

4.4 RPL Efficiency Evaluation

We present next an evaluation of RPL in terms of reliability, end-to-end delay, energy consumption, and induced overhead. We also investigate the dynamics of the routing topology when the DODAG is constructed using as routing metrics: hop count, ETX, and LQI.

4.4.1 Reliability

We first evaluated the reliability of RPL, which we measured through the end-to-end packet delivery ratio (PDR). The PDR is computed as the ratio of the number of packets received by the border router, and transmitted by each node.

Figure 4.3 illustrates the Complementary Cumulative Distribution Function (CCDF) of the end-to-end PDR for all the nodes. Not surprisingly, hop count exhibits the worst PDR. It tends to privilege long and potentially bad radio links to forward the packets. Because LQI and ETX consider the quality of the radio link, they present better performance.

We illustrated the PDR with boxplots in Figure 4.4, by grouping the source nodes according to their geographic distance from the border router. We can extract a strong correlation between the distance and the PDR, especially in the case of hop count: the farther a node is, the lower its PDR is. Indeed, more nodes have to forward the packets, creating more packet losses. We can also notice that for nodes that are not at one hop from the sink (situated farther than 300m), some of them present a very bad PDR (under 40%). Indeed, there might be 1 or 2 nodes that have a hard time finding a good radio link to their preferred parent.

Both LQI and ETX seem to be good link quality indicators. However, when the distance from the border router increases, the LQI accuracy decreases, having the worst case close to the hop count. A node receives less packets, so it is more difficult for LQI to properly estimate the quality of the link.

4.4.2 End-to-end Delay

We evaluated the end-to-end delay, i.e., the time between the packet generation at the source, and its reception by the border router (considered only for delivered packets). Figure 4.5 presents

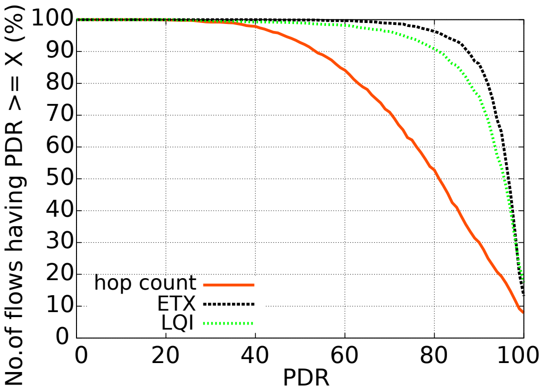


Figure 4.3: Complementary CDF of the end-to-end PDR

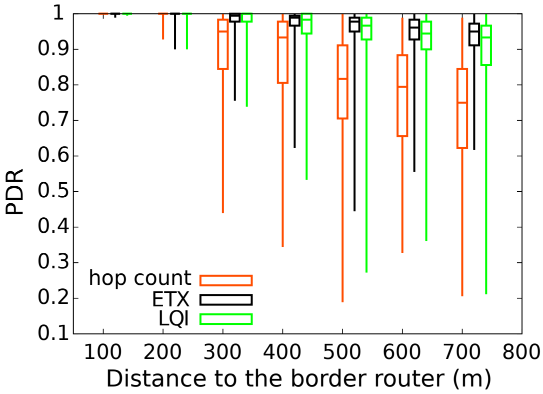


Figure 4.4: PDR vs. geographic distance to the border router

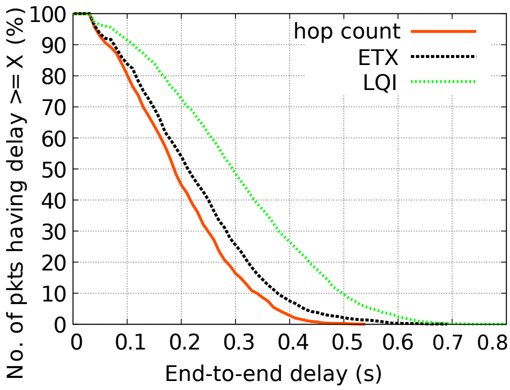


Figure 4.5: Complementary CDF of the end-to-end delay

the results as a Complementary Cumulative Distribution Function (CCDF).

We can observe that hop count presents the best end-to-end delay. We saw that most of the packets are dropped, particularly from nodes far from the border router. Since the delay is only computed for the received packets, the nodes closer to the border router (with shorter paths) are over-represented in the end-to-end delay result.

LQI has the worst end-to-end delay. This is due to the fact that it has a bad PDR for the nodes situated further away from the border router. Those nodes will have more retransmissions, which negatively impacts the delay.

4.4.3 Energy Efficiency

We also studied the energy efficiency of RPL. Since the MAC protocol that we used here (IEEE 802.15.4 beacon-less) keeps the radio always on, we estimated the energy consumption with the number of transmitted packets. If we use a duty cycle approach, most of the energy consumption is drained by packet transmissions.

Figure 4.6 presents the boxplot for the packets transmitted by the nodes, in function of their distance to the border router. Since we are interested in the total energy consumed by the nodes, the graph also accounts for the retransmissions at the MAC layer.

We can notice that the nodes closer to the border router have more packets to forward, and consequently consume more energy. It seems that the nodes that are at 200m from the border router consume more energy than the nodes closer (at 100m). However these nodes are still at

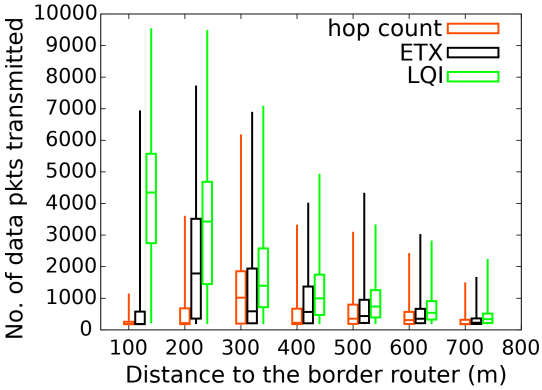


Figure 4.6: Number of packets transmitted vs. geographic distance to the border router

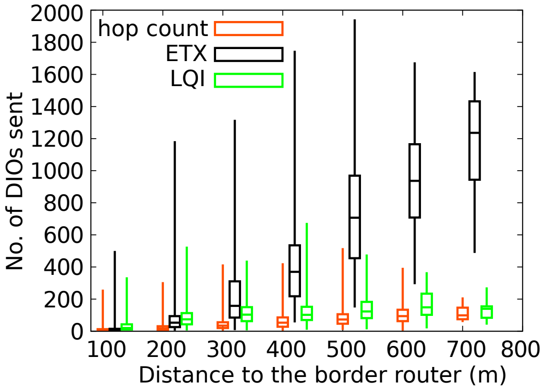


Figure 4.7: Number of DIO transmitted vs. geographic distance to the border router

one hop distance to the root, so it is likely for them to be chosen as preferred parent in order to minimize the number of hops (in the case of hop count) or the cumulative link quality (in the case of ETX).

Also, hop count balances well the load among the different nodes. On the contrary, with LQI, a non negligible portion of the nodes consume most of the resources. This could be problematic for the network lifetime. While the MAC layer can minimize the energy consumption by using a low duty cycle, the network layer does it through mechanisms ensuring efficient route selection that avoids bottlenecks, congestion, etc. RPL should provide such a mechanism to avoid this unfair energy consumption.

4.4.4 Overhead

Finally, we studied the routing overhead induced by the control packets (in our case, just by the `DIO`, since `DAO` messages are deactivated, and `DIS` packets are used only in the case of a local repair). Figure 4.7 presents the boxplot for the `DIO` packets transmitted by the nodes through the whole simulation period, in function of their distance to the border router.

We can notice that ETX sends the largest number of control packets among the used routing metrics. This can be explained by the fact that a small variation in the link quality can determine the Trickle timer to reset more often, sending `DIO` more frequently, and creating a larger overhead.

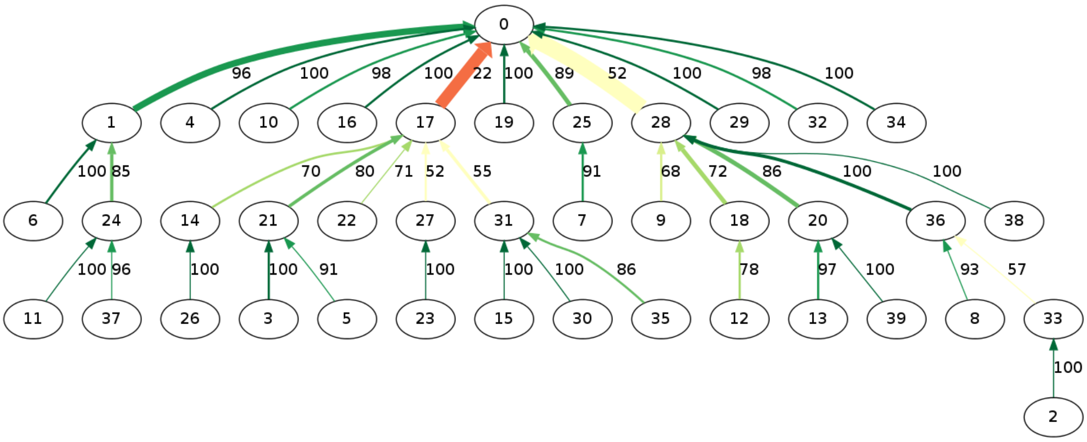
Because of the cumulative ETX variations, we can also observe that the nodes situated further away from the border router tend to send more control packets. When one node resets its Trickle timer because an inconsistency was detected in the network, all the nodes in its sub-DODAG have to update their information. This can lead to them also resetting their Trickle timer.

Let us take a closer look at this phenomenon. We have illustrated in Figure 4.8 the routing topology constructed during the simulation, when ETX and hop count are used as routing metrics. Even though the simulated topology is the same, the resulted DODAGs are very different. The objective of hop count is to create shortest paths, hence the DODAG is more compact, with a depth of 4 (Figure 4.8a). On the contrary, ETX constructs longer paths, in order to find the best links. This results in a more spread topology, with a DODAG depth of 6 (Figure 4.8b).

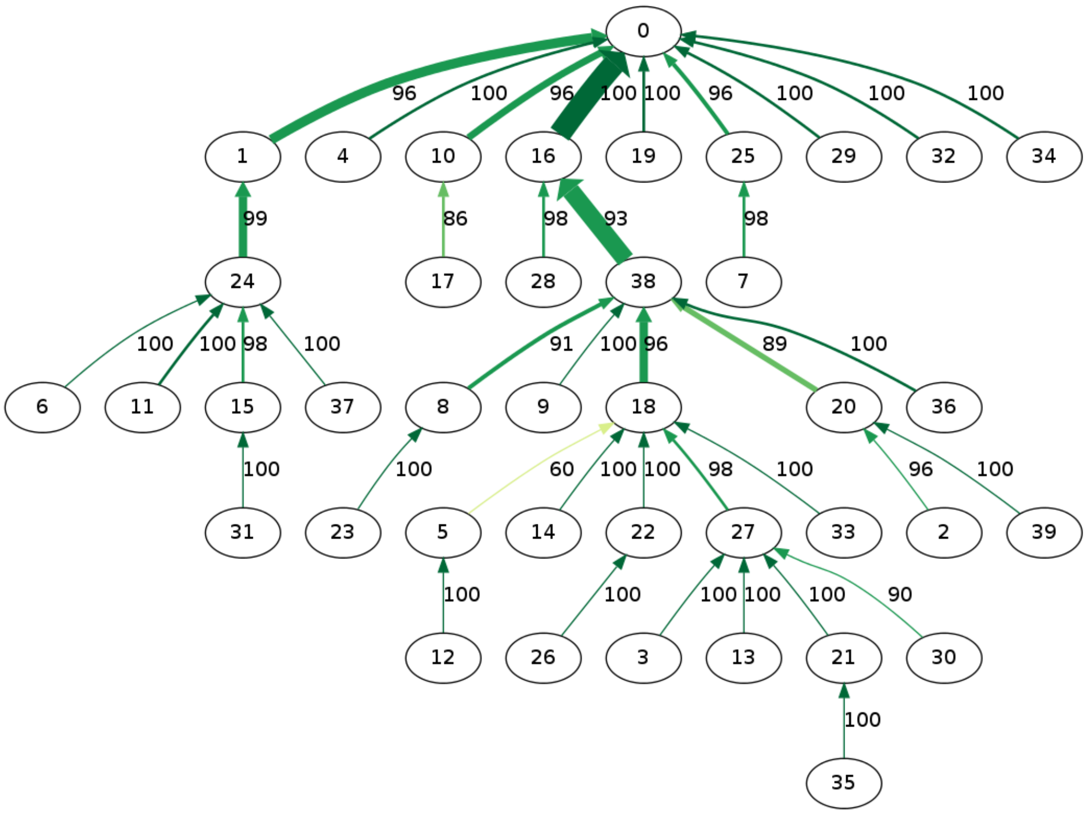
Since the DODAG has a greater depth when ETX is used as a routing metric, implicitly the nodes further away will send more `DIO`. On the contrary, when hop count is used, the increase in `DIO` control packets with the distance from the border router is less visible.

4.5 DODAG Stability Evaluation

During the evaluation of the efficiency of RPL under different metrics, we found out that ETX induces a lot of overhead in the network, because of the frequent resets of the Trickle timer. In



(a) Using hop count as a routing metric



(b) Using ETX as a routing metric

Figure 4.8: The state of the DODAG at the end of the simulation
(for a better visibility, we show here the results for a network with only 40 nodes):

- the width of the arrow expresses the quantity of traffic forwarded by the link: the wider the arrow, the larger the traffic;
- the color of the arrow expresses the quality of the link: the greener (the more red) it is, the better (the worse) the PDR of the link is;
- the label of the edge represents the PDR×100 of the corresponding link.

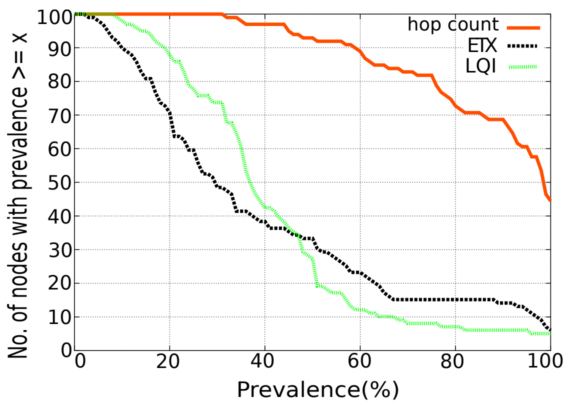


Figure 4.9: Complementary CDF of the end-to-end route prevalence

order to better understand this phenomenon, we focused on investigating the routing dynamics of the DODAG. Even if the topology is stable, the radio conditions are stochastic and may impact the behavior of RPL.

4.5.1 Route Prevalence and Persistence

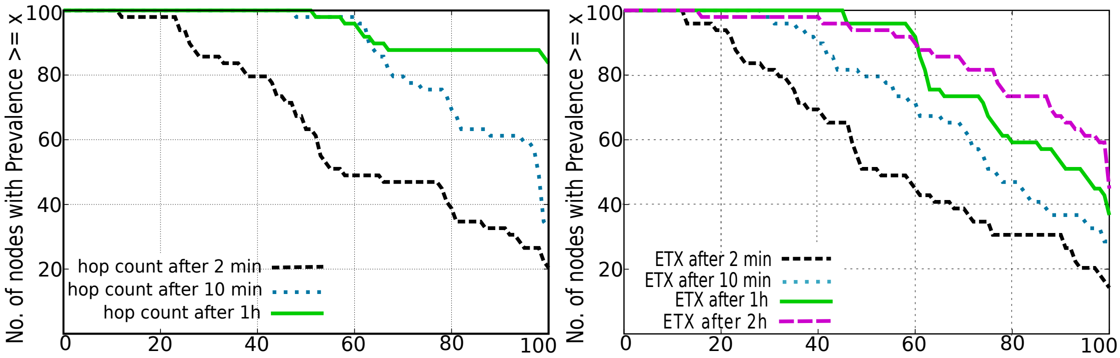
We first investigated the routing stability, by measuring:

- the route prevalence: the average ratio of time during which we observe the same route (i.e., the principal route) [Pax96]. It is computed as the ratio between the number of times the principal route was used, and the number of times all routes have been used;
- the route persistence: the average time before a route changes (to detect route flapping) [Pax96]. It is computed as the ratio of the total time the principal route was used, and the number of times it was used.

Route Prevalence

Figure 4.9 illustrates the CCDF of the route prevalence. The DODAG is very stable when we use hop count, although it performs poorly, as highlighted previously: 45% of the nodes use one single route to send the packets to the border router. On the contrary, ETX and LQI exhibit many changes. For instance, one third of the nodes use the principal route only 20% of the network lifetime with ETX. Since a parent change generates a large overhead in RPL, these metrics waste resource energy.

We suspected that this instability problem comes from the initialization phase. Thus, we measured the route prevalence over a period of 5 hours, while removing the bootstrap period, for which we tried several values (Figure 4.10). We can observe in Figure 4.10a that if we remove the first hour of simulation when using hop count, almost 90% of the nodes use the principal route. On the other hand, when using ETX, the network is very dynamic (Figure 4.10b). Even if we remove the first 2 hours, still only 50% of the nodes direct their traffic on the principal route. In other words, RPL seems to not converge to a stable set of routes, even when nodes are static, the traffic is CBR, and the radio conditions remain unchanged. In order for RPL to converge, it needs an ideal estimator of the link quality, which is very difficult to compute, since it must be reactive to persistent changes, yet stable in front of transient (short-term) variations. However, in the absence of such an estimator, the mechanisms of RPL should be able to handle a biased estimator, and this is not the case.



(a) Hop Count (b) ETX

Figure 4.10: CCDF of the end-to-end route prevalence after initialization

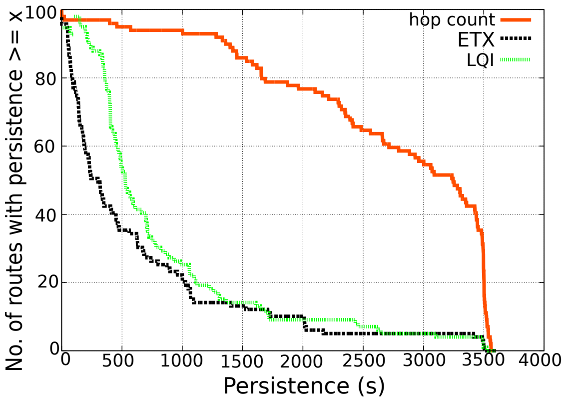


Figure 4.11: Complementary CDF of the end-to-end route persistence

Route Persistence

Figure 4.11 illustrates the route persistence. We can observe that in the case of ETX, 50% of the nodes use the principal route an average of 500 seconds (over a simulation run of 1 hour), which suggests that the routes are short-lived. Only hop count exhibits a stable behavior. But it does not constitute an accurate solution, since it performs very badly.

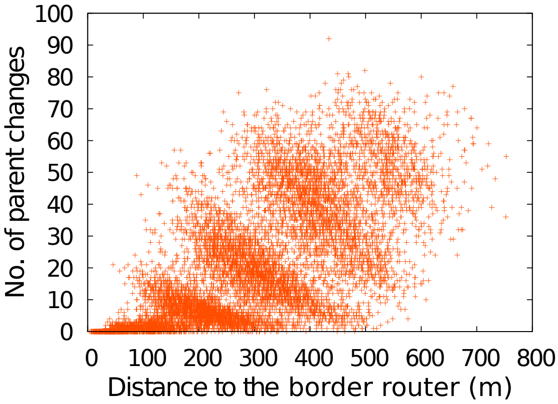
4.5.2 Stability of the Preferred Parent

We also measured the stability of the DODAG by calculating the number of times each node changes its preferred parent. Figure 4.12 illustrates the number of parent changes for each node against the distance of the nodes to the border router. Results presented here concern a network of 500 nodes, to highlight clearly the effects of each routing metric. However, we verified that we obtain similar results with lower node density. We also maintained the same duration of the simulation (3600 seconds).

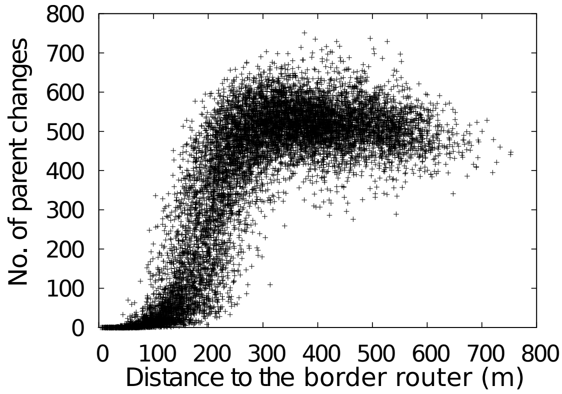
The reader may remark that the y-scale is different for each graph. While a node changes at most 800 times its preferred parent with ETX (Figure 4.12b), the number of parent changes is more reduced with hop count (Figure 4.12a) and LQI (Figure 4.12c).

While with ETX the network is highly dynamic, LQI has a different behavior (Figure 4.12c): the number of parent changes increases linearly with the distance. Thus, LQI is quite scalable concerning the border router distance. However, it keeps on generating useless control packets even if the radio topology and conditions do not change during the simulation.

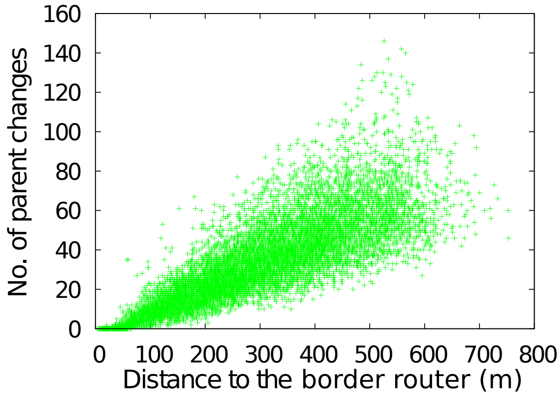
When hop count is used as a metric (Figure 4.12a), we denote the appearance of a *step*



(a) Hop Count



(b) ETX



(c) LQI

Figure 4.12: Number of parent changes in the DODAG

pattern. Each step represents the hop distance to the border router.

Let us consider Figure 4.13 to explain this pattern. A, B, D and E are sufficiently *close* (signal distance) to the root and choose it as preferred parent. Node C may choose either A or B as parent. On the contrary, F is farther away, and only the node D is a radio neighbor: the overlap between the *1-hop region* and the *radio range* is much smaller than for the node C. F will consequently not change its preferred parent: its choice is limited. This phenomenon explains the *step* pattern.

Microscopic View

Let us analyze these parent changes in a microscopic manner. We took as an example a topology where the sink is situated in the middle. More precisely, at the coordinates (377, 301). Then, we plotted the spatial distribution of the number of parent changes in the DODAG, with each routing metric.

The results, presented in Figure 4.14, offer a better view on the previously described phenomenon. First, we can remark that the y-scale is still different for each graph. Second, the hop count metric manages to keep a low number of parent changes. Once it has chosen a preferred parent, it will keep it, no matter the link quality, until the MAC layer announces that it is not reachable anymore (i.e., a maximum number of packets has been send without receiving an acknowledgement).

Both LQI and ETX use an estimator of the link quality to choose their preferred parent. We

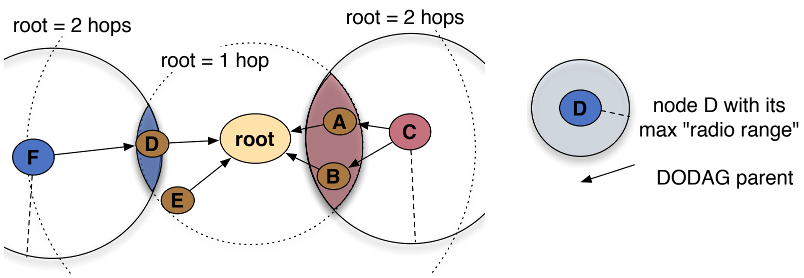


Figure 4.13: Preferred Parent choice (redundancy) for nodes at a different distance from the border router

can observe that the number of parent changes increases with the distance from the sink. While this increase looks steady in the case of LQI (a little exacerbated for the nodes in the corners), it is more dramatic for ETX. Moreover, these instabilities due to preferred parent changes will be amplified when the sink is situated in a corner, since the DODAG has a greater depth.

4.6 Conclusion

In this chapter, we made a thorough evaluation of the performance of RPL, and we highlighted the existence of a tradeoff between network stability and efficiency.

Hop count does not use any link quality estimators when choosing the preferred parent, and in consequence obtains the worst reliability. Since the only information that it requires is the number of hops on the route, it is the simplest metric to compute and hence, it does not induce too much overhead.

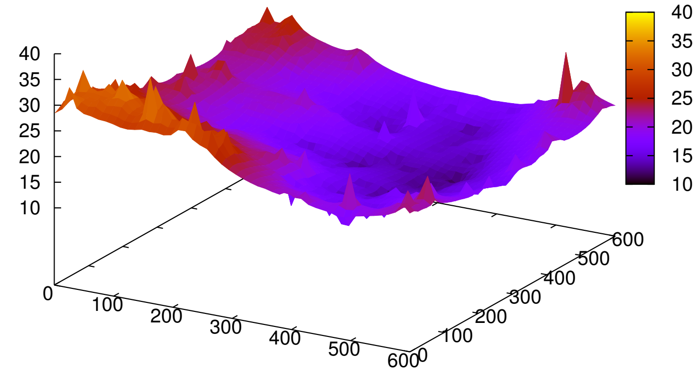
LQI also has a low overhead, but offers larger end-to-end delays, and a non negligible portion of the nodes consumes most of the resources.

On the contrary, ETX uses statistics about the PDR of the link, and thus manages to choose the best available paths. As a result, RPL has the best end-to-end reliability when constructing the DODAG with this metric, even for the nodes situated far away from the border router. As ETX is always searching for the best instantaneous link quality, a node often changes its preferred parent, inducing instability in the network.

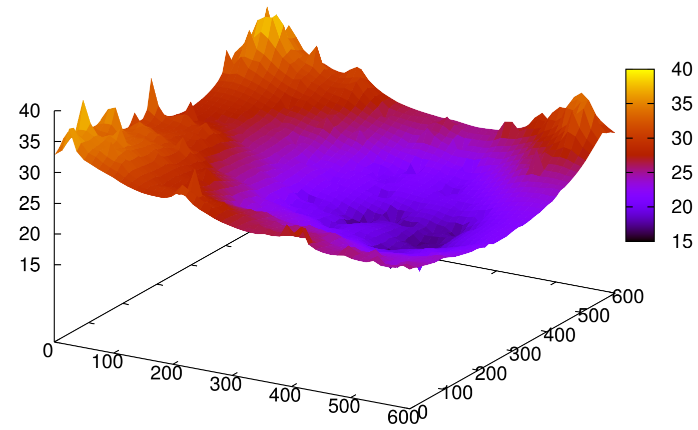
We investigated the route prevalence (the average ratio of time during which we observe the principal route) and found out that with ETX, one third of the nodes use the principal route only 20% of the network lifetime. The network remains highly dynamic, even if we remove the bootstrap period. It will be interesting to see also the distribution of the prevalence of these routes. If there are only a few routes with close cumulative ETX, we could propose an algorithm for consistency in the choosing of a route. However, the more different the routes are, the more difficult it will be to make such a decision.

While this instability problem was highlighted only by simulations, it will surely remain unsolved also in experimental/real conditions. A fortiori, the presence of asymmetrical links and a more fast varying radio channel will amplify the problem rather than solving it.

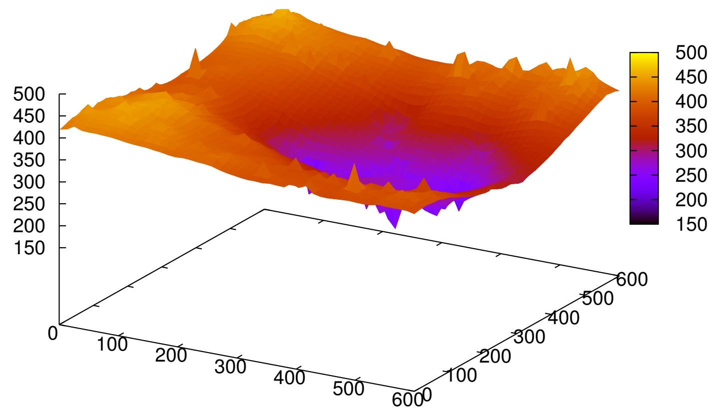
In order to reduce these instabilities, and to maximize the network lifetime, we propose in the next chapter a new routing metric.



(a) Hop Count



(b) LQI



(c) ETX

Figure 4.14: Spatial distribution of the number of parent changes in the DODAG Sink situated at (377, 301)

Chapter 5

Maximizing the Lifetime of WSNs through Energy Balancing Routing

The lifetime of Wireless Sensor Networks is very limited, since most of their devices are energy constrained. MAC protocols that offer a low duty cycle (e.g., IEEE 802.15.4) became essential to extend network lifetime: a node may periodically turn-off its radio to save energy. In parallel, energy efficient routing has been researched on with three major objectives:

1. minimize control packet exchange;
2. avoid the weakest nodes (i.e., with a low residual energy);
3. use reliable and energy efficient links.

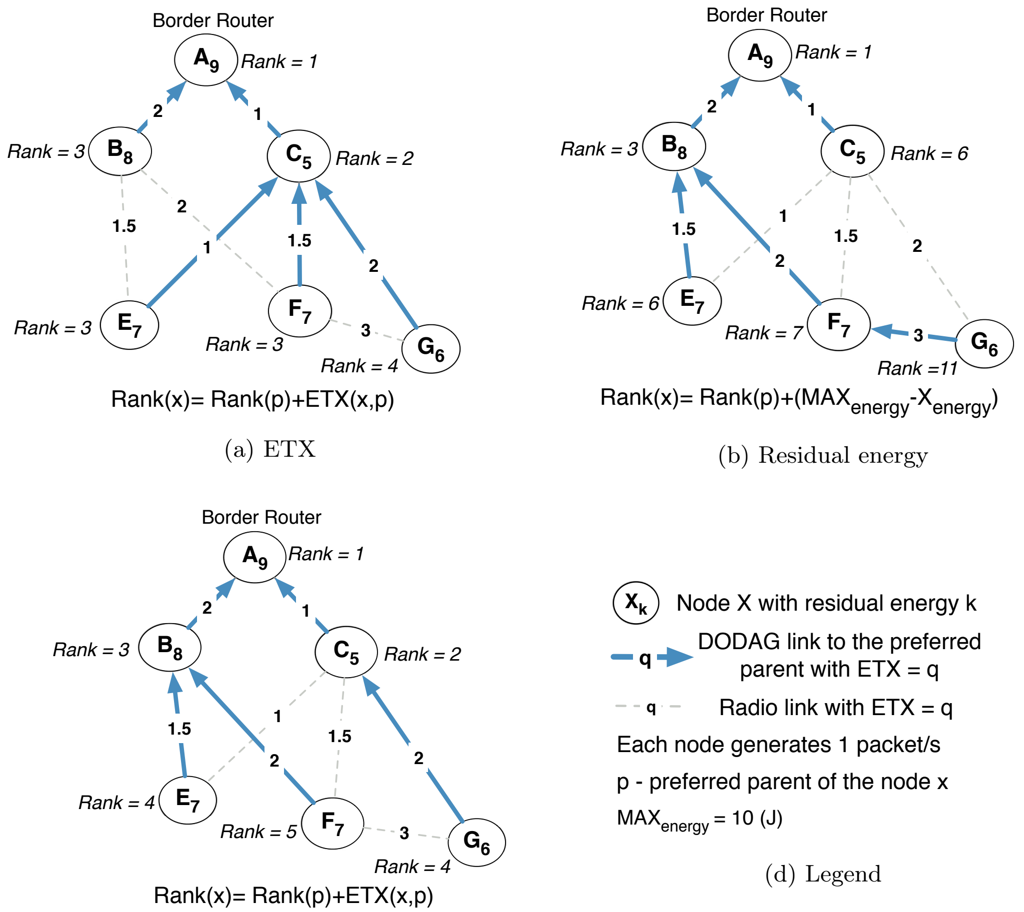
We aim here at proposing an energy efficient routing layer by incorporating all these 3 objectives. We take a novel approach by **maximizing the lifetime of the most constrained node**, rather than minimizing the average energy consumption. We identify the bottlenecks of the network in terms of energy (i.e., the nodes that have the least residual energy), and we route the packets in order to maximize their lifetime. We thus obtain energy balanced paths.

In consequence, we propose a new routing metric that estimates the *Expected Lifetime* of a node, according to its residual energy, the link quality to its neighbors, and the current traffic conditions. By appropriately constructing a network topology based on this metric, we are able to improve globally the network lifetime.

Contribution

This chapter presents the following contributions:

1. We propose the *Expected Lifetime (ELT)* routing metric (i.e., how much time a node has to live before it runs out of energy), and show how to estimate it for each node;
2. We present an algorithm to compute a path metric, based on this node metric, which will globally maximize network lifetime;
3. We apply this approach to RPL (the *de facto* routing standard in the Internet of Things), and thoroughly evaluate its behavior through simulations.



(a) ETX

(b) Residual energy

(c) ELT

(d) Legend

Figure 5.1: DODAG construction using different routing metrics

5.1 Problem Statement

Two main approaches to save energy exist in the literature. The first one minimizes the global energy consumption. The ETX metric for instance, may take into account the link reliability to construct only energy-efficient routes [Cou+05]. However, the nodes with the best links will be chosen uppermost to route packets. They will deplete their energy faster.

Let us consider the topology in Figure 5.1a where the RPL DODAG is constructed based on the ETX metric [Vas+12]. E may choose either B or C as next hop. C is the most accurate choice, since it presents the lowest cumulative ETX toward the border router. However, if all the nodes generate the same amount of traffic, B should be preferred to balance the energy consumption.

The second approach gives higher priority to the nodes with a large residual energy [Kam+12]. Still, these nodes, with possibly bad links, will receive most of the traffic and will consequently run out of energy faster. Clearly, we should not have a routing strategy that concentrates most of the traffic on a small collection of nodes.

Let us take a look at the topology depicted in Figure 5.1b where the RPL DODAG is constructed based on the residual energy. We can observe that G can choose either F or C as a next hop. Because the residual energy of F is greater, it will choose it as a parent, even though the corresponding link quality is very low (ETX=3). This will lead to the quick battery depletion of node G. C would be a more appropriate choice since it would result in a more energy balanced topology. Indeed, B will not have to relay the whole traffic of the network.

None of these approaches manages to create a routing topology that will maximize the lifetime of a WSN. We need a routing metric that considers the quality of the links while balancing the load in function of the available energy of the nodes on that path. We are talking here about energy balancing: constructing paths that spend the same energy.

If we take a look at Figure 5.1c we can see an example of an energy-balanced routing topology. A node selects its preferred parent so that it maximizes the lifetime of the most constrained nodes, without becoming itself the new constrained node.

In short, the routing metric should satisfy the following properties:

- capture link quality variations (dynamic);
- maximize the reliability;
- minimize the energy consumption of the bottlenecks (i.e., the nodes that consume the most energy). Balancing the energy should be preferred in order to prolong the network lifetime.

5.2 The Expected Lifetime

We propose here a new routing metric that estimates the *Expected Lifetime* of a node, according to the residual energy, the link quality, and the current traffic conditions. By appropriately constructing a network topology based on this metric, we are able to improve globally the network lifetime.

5.2.1 Assumptions

We consider the network lifetime as the time before the first node runs out of energy, since it is the most frequent definition [Kam+12; Yoo+10; KDB13; Lee+14]. Moreover, a node should choose to send its traffic on the least energy-constrained path. Hence, if a node runs out of energy it will most surely disconnect the network, otherwise, its neighbors would not have chosen it as a parent.

We have a WSN with a periodic traffic pattern, since it is the most common one (Table 1.1): all the sensors periodically report their measurements to a border router. We assume the energy consumption is dominated by the data packets, and leave aside the control packets.

We also consider that the energy consumed to receive a packet can be neglected. Indeed, in IEEE 802.15.4-2006 for example, a node transmits data packets to its coordinator (Figure 5.2). The transmitter is able to turn off its radio during the backoff and the idle time. Consequently, the quantity of energy consumed by a transmitter is roughly related to the number of transmissions. On the contrary, the coordinator (i.e., the receiver) has to stay awake during the whole active period. Besides, the power of the radio chipset is often the same, no matter if a packet is received or not (e.g., Atmel AT86RF231). Consequently, a coordinator does not spend more energy to receive a packet: it has anyway to stay awake (idle listening) during the whole active period.

5.2.2 The Expected Lifetime (ELT) of a Node

We propose here the *Expected Lifetime* (ELT) routing metric. Instead of minimizing the sum of energy, or considering only the residual energy, ELT aims at maximizing the lifetime of the most constrained nodes, denoted *bottlenecks*.

ELT estimates the expected lifetime, i.e., the time before a node dies if it keeps on forwarding the same quantity of traffic. ELT helps to quantify the impact of a routing decision on the bottlenecks.

To compute its ELT, a node N (cf. notation in Table 5.1):

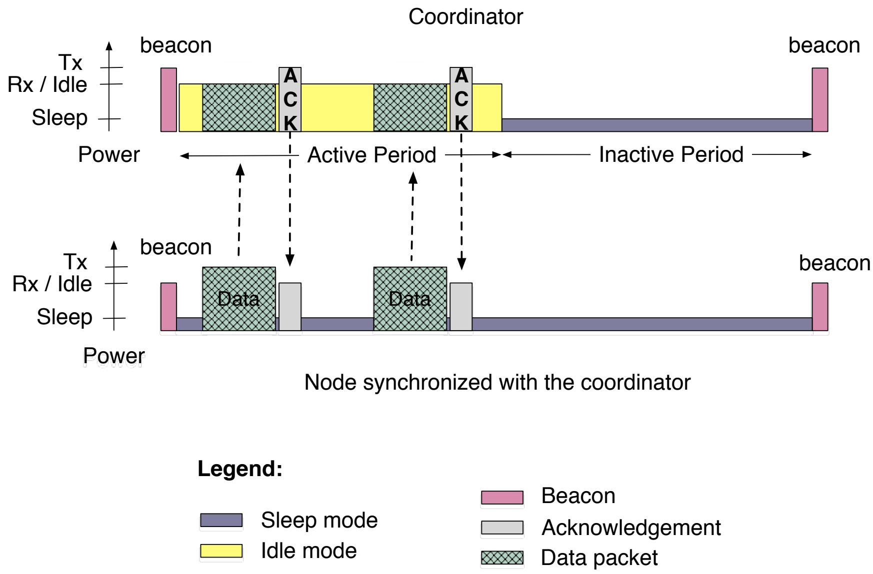


Figure 5.2: Communication between a node and its parent (i.e., a coordinator) in IEEE 802.15.4

1. estimates the total traffic that it has to transmit by adding the traffic that it generates to the incoming traffic from its children (i.e., the layer-3 load):

$$\mathbf{T_N} = T_{gen}(N) + \sum_{i \in Children(N)} T_i \tag{5.1}$$

2. multiplies the traffic that it has to transmit by the average number of retransmissions, given by the link reliability to its preferred parent (ETX(N, P_N)): the more retransmissions are needed, the more energy is consumed:

$$T_N \times \mathbf{ETX(N, P_N)} \tag{5.2}$$

3. computes the ratio of time during which it uses the medium for transmissions, by including the rate at which the data is sent:

$$\frac{T_N \times \text{ETX}(N, P_N)}{\mathbf{DATA\_RATE}} \tag{5.3}$$

4. computes the energy spent to transmit all the traffic by multiplying the ratio of time during which it uses the medium, with the transmission power of its radio:

$$\frac{T_N \times \text{ETX}(N, P_N)}{\text{DATA\_RATE}} \times \mathbf{P_{TX}(N)} \tag{5.4}$$

5. finally, N computes its remaining lifetime as the ratio between its residual energy, and the energy spent to transmit its traffic:

$$ELT(N) = \frac{\mathbf{E_{res}(N)}}{T_N \times \frac{\text{ETX}(N, P_N)}{\text{DATA\_RATE}} \times P_{TX}(N)} \tag{5.5}$$

If we compare our proposal (ELT) with the other metrics presented in Table 2.2 we can notice that it manages to overcome their limits: it uses a passive measurement technique to estimate the link quality, and it considers both the number of retransmissions and the residual energy. Since ETX reduces the number of retransmission on the path, it indirectly reduces also the delay. By using ETX in our metric, we manage to partially reduce the delay, as well.

Table 5.1: Notation used for the ELT metric

Notation	Meaning
ELT(X)	Expected lifetime of X
$\boldsymbol{E_{res}(X)}$	Residual energy of X (in Joule)
$\boldsymbol{P_{TX}(X)}$	Radio power in transmission mode (in Watt or Joule/s)
ETX(A,B)	ETX of the link $A \rightarrow B$
$\boldsymbol{T_X}$	Throughput (bits/s) of X
$\boldsymbol{T_{gen}(X)}$	Traffic generated by X
Children(X)	Children set of node X
Parents(X)	Parents set of node X
$\boldsymbol{B_X}$	Bottleneck of the path through node X
$\boldsymbol{P_X}$	Preferred parent of the node X
DATA_RATE	The rate at which the data is sent (bits/s); All nodes transmit at the same rate

5.3 The Expected Lifetime - Application to RPL

We aim at improving the lifetime of the network. To this end, we need to **minimize the energy consumption** of the **most constrained node**, denoted as the **bottleneck**. We consequently have a *min-max* objective. This problem may also be translated into its dual problem: maximizing the lifetime of the most constrained node.

Since we want to maximize the network lifetime, we need to focus our decision on the energy bottleneck i.e., the node that is most likely to be the first one to die. Thus, the weight of a path is the minimum ELT between all the traversed nodes. For example, in Figure 5.3, the bottleneck of the path from G to A is the node C: it has the lowest ELT of the path.

We will now explore how we may implement this *min path metric* in a gradient routing scheme. To be used with RPL a node needs to:

1. estimate its own impact on the ELT of the bottleneck of a path;
2. send the information about the bottleneck along the path in the control packets (i.e., `DIO`) in a compact manner;
3. choose its preferred parent (i.e., the next hop) so that it creates energy-balanced paths;
4. construct a loop-free topology by paying attention to the computation of its rank: it must be strictly and monotonically increasing from the sink, toward the leaves.

We will now address each of these challenges in the next subsections.

5.3.1 Estimation of the ELT of a Bottleneck

Let us consider that a node N wants to join the DODAG. Since the bottleneck is most likely to be the first node to die, the new node has to estimate the impact of its own packets on the lifetime of the bottleneck.

If we take a look at Equation 5.5, we can notice that only the throughput (T_N) is dependent on the traffic injected by the new node. Hence, in order to estimate how N influences the lifetime of the bottleneck, we add the traffic of N to the current throughput of the bottleneck. If B is the bottleneck, then:

$$\mathbf{new\_T_B} = T_B + T_N \tag{5.6}$$

N can now estimate its impact on the lifetime of B:

$$ELT(B) = \frac{E_{res}(B)}{\mathbf{new\_T_B} \times \frac{\text{ETX}(B,P_B)}{\text{DATA\_RATE}} \times P_{TX}(B)} \tag{5.7}$$

Once N knows its impact on the ELT of a bottleneck, it can use this information to create energy balanced paths. But first, let us see how the different information about the bottleneck (throughput, residual energy, etc.) are advertised in the network.

5.3.2 Compact `DIO` Advertisement

In order to save memory and energy, we need to compress the information about the bottleneck, i.e., we have to minimize the number of fields to insert in the `DIO`. We can adopt the same reasoning as for Equation 5.6. The only component that is dependent on the traffic added by a new node is the throughput. Hence, all the other elements can be captured into a single constant, called $B\_constant$.

In consequence, a `DIO` will contain the following information:

- **id:** the bottleneck id: IPv6 address (16 bytes), 6LowPAN address (4 bytes) or IEEE 802.15.4 short address (2 bytes);
- **existing_traffic:** the traffic forwarded by the bottleneck (normalized): T_B (1 byte);
- **B_const:** the normalized value of the bottleneck constant (2 bytes) computed as:

$$B\_const = \frac{E_{res}(B)}{\frac{ETX(B,P_B)}{\text{DATA\_RATE}} \times P_{TX}(B)} \tag{5.8}$$

Thus, besides the configuration parameters, a `DIO` will contain in its `DAG Metric Container` the three variables: *id*, *existing_traffic* and *B_const*. Indeed, this information is sufficient for a new node N to accurately estimate the impact its traffic will have on the bottleneck B:

$$ELT(B) = \frac{B\_const}{existing\_traffic + T_N} \tag{5.9}$$

where T_N is the traffic injected by the new node on the path having the bottleneck B.

5.3.3 Preferred Parent Selection

When choosing its preferred parent, a node must consider both its own lifetime and the lifetime of the bottleneck, in order to estimate which of them becomes the new bottleneck.

We consequently propose the Algorithm 1 for a node to select its preferred parent (following the notation from Table 5.1). For each possible parent (i.e., a neighbor advertising a rank smaller than itself) a node N:

1. computes its own lifetime when choosing this parent (line 2);
2. computes the lifetime of the bottleneck on the path advertised by this parent, by adding its own traffic to the throughput of the bottleneck (line 3);
3. saves the minimum value of the lifetime among both, in case it became itself the new bottleneck of that path (line 4).

Finally, the parent which presents the largest minimum lifetime is selected as the preferred parent (line 6). The node then computes the new bottleneck of the path and updates the corresponding information in its `DIO`.

Algorithm 1: Preferred parent selection

Data: N, $Parents(N)$
Result: preferred_parent of N
for $P \in Parent(N)$ **do**
    $elt_N = ELT(N)$;
    $elt_B = ELT(B_P)$;
    $Path_P(B_P) = \min\{elt_N, elt_B\}$;
end
preferred_parent $= P$ such that $Path_P(B_P) = \max\limits_{i \in P(N)} \{Path_i(B_P)\}$

5.3.4 Loop Freeness and Rank Computation

Sobrinho [Sob03] has proved that for a distance vector protocol like RPL to be loop free, the routing metric must be strictly monotonic. Strict monotonicity implies that the weight of the path does not decrease (if the metric is monotonically increasing) or increase (if the metric is monotonically decreasing) when prefixed or appended by another path.

In our case, the weight of a path represents the minimum ELT on that path, i.e., the ELT of the bottleneck. Since we want to maximize the network lifetime, the weight of a path should not decrease when appended/prefixed with another path.

Let the path p be prefixed with the path q. The ELT of the path p may:

1. remain stable if p keeps on presenting the lowest ELT;
2. be smaller than the ELT of p if the ELT of q is less than or equal to that of p.

In other words, ELT is not strictly monotonic, and is susceptible to create loops.

RPL specifies that to obtain a loop free DODAG, the rank of the nodes must be strictly monotonically increasing from the border router towards the leaves. Still, its exact calculation is left to the objective function. To fully exploit the flexibility of RPL, we propose here to separate the metric that we use to construct the DODAG from the metric used to compute the rank.

In consequence, we propose that a node computes its rank by adding a step value called *Rank_increase*, to the rank of its preferred parent. This value is derived from the ETX of the link to its preferred parent:

$$\begin{aligned} Rank(N) &= Rank(P_N) + Rank\_increase \\ Rank\_increase &= ETX(N, P_N) \times \texttt{MinHopRankIncrease} \end{aligned} \tag{5.10}$$

where P_N is the preferred parent of N and `MinHopRankIncrease` the RPL parameter [Win+12].

We chose to use ETX and not a simple metric like hop count, in order to reflect the link quality in the rank. Also, this way, we increase the number of parents that a node can have. Indeed, with hop count, the rank increases linearly, which reduces the number of neighbors that could be used as backup parents.

Clearly, such metric is monotonic, guaranteeing loop-freeness. Indeed, the weight of the path is the cumulative ETX on that path. When prefixed or appended with another path, its weight cannot decrease.

RPL forbids a node to consider as next hop a neighbor with bigger rank than itself. Thus, we ensure from the begging (i.e., bootstrap) the formation of a loop-free topology. Also, by pairing the rank and the choice of the preferred parent we keep the maximum lifetime.

5.3.5 Illustration

If we take a look at the example in Figure 5.3, G may choose as preferred parent the node D or F. If it chooses the path with the largest ELT for the bottleneck (through node D), it will

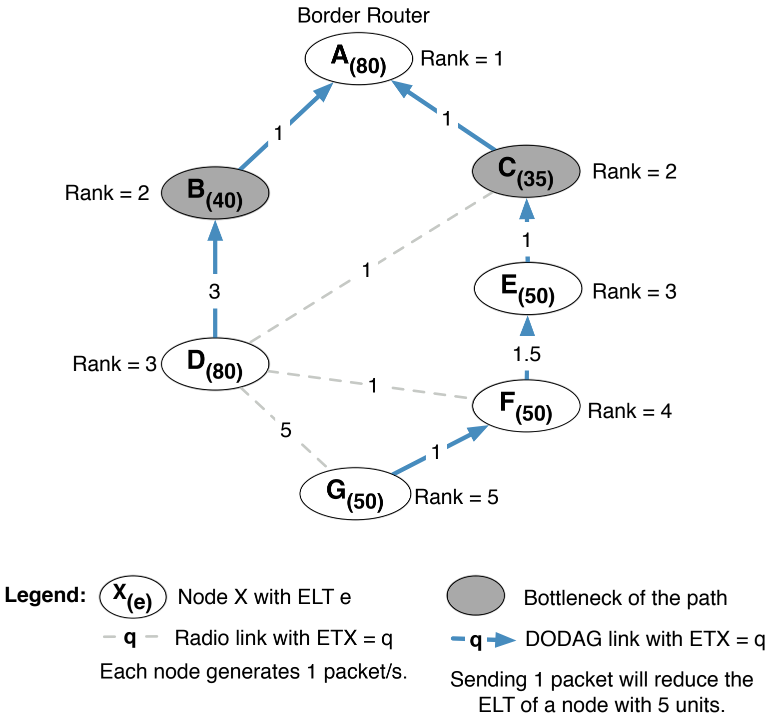


Figure 5.3: Path selection with ELT ($Rank\_increase = 1$)

become itself the new bottleneck: the quality of the link between G and D is very bad (ETX=5), so it will need a lot of retransmissions for a packet to successfully arrive at D. Indeed, the ELT of B will drop to 35, while G will become the new bottleneck with an ELT of 25. On the other hand, if G chooses F as a preferred parent, it will have a small impact on both the ELT of the bottleneck (new ELT of C will be 30) and of its own (ELT of G will drop to 45).

Finally, G will choose F, since it maximizes the minimum ELT between all the nodes in the network. G computes its rank as the rank of its preferred parent plus the constant $Rank\_increase$ (in our case, 1), and starts advertising information about the new bottleneck of the path. The new bottleneck is the minimum value between its ELT and the ELT of the bottleneck of the path from its preferred parent to the border router (in this example, C remains the bottleneck of the path).

We can notice that in this example, the choice of the preferred parent would be the same in case ETX will be used as the routing metric. On the contrary, D will not choose the same parent if it uses ETX. It would choose C (ETX=1) rather than B (ETX=3).

5.3.6 Proof of Lifetime Maximization

Let N be a node in a WSN that has to choose its preferred parent between P and Q, where P offers the optimal path (largest ELT). Let ELT(N_P) (respectively ELT(N_Q)) be the expected lifetime of node N if P (respectively Q) would be its preferred parent. Let us provide here a proof by contradiction.

P offers the optimal path means that after N chose its preferred parent, the ELT of the new bottleneck through P is greater than the bottleneck through Q and greater than ELT(N_Q). Since the bottleneck through P can be either the old bottleneck advertised by P (B_P), or N (if its ELT is smaller than ELT(B_P)), we can distinguish two cases:

1. $new\_B_P = B_P \Rightarrow$

$$\text{ELT}(B_P) > \text{ELT}(B_Q) \ \& \ \text{ELT}(B_P) > \text{ELT}(N_Q) \tag{5.11}$$

2. $new\_B_P = N \Rightarrow$

$$\text{ELT}(N_P) > \text{ELT}(B_Q) \ \& \ \text{ELT}(N_P) > \text{ELT}(N_Q) \tag{5.12}$$

Let us suppose now that instead of choosing the optimal path, N chooses the path through Q. This means that:

$$\text{ELT}(new\_B_Q) > \text{ELT}(B_P) \ \& \ \text{ELT}(new\_B_Q) > \text{ELT}(N_P) \tag{5.13}$$

Like before, we can distinguish two cases:

1. $new\_B_Q = B_Q \Rightarrow$

$$\text{ELT}(B_Q) > \text{ELT}(B_P) \ \& \ \text{ELT}(B_Q) > \text{ELT}(N_P) \tag{5.14}$$

2. $new\_B_Q = N \Rightarrow$

$$\text{ELT}(N_Q) > \text{ELT}(B_P) \ \& \ \text{ELT}(N_Q) > \text{ELT}(N_P) \tag{5.15}$$

Both cases lead to a contradiction. In conclusion, a node will always choose as preferred parent the one that maximizes the lifetime of the bottleneck and therefore, of the network.

5.4 Simulation Results

We used the WSNet simulator, an event-driven simulator dedicated to WSNs, which has been extensively evaluated [HCG08]. The results are averaged over 20 simulations with different random topologies, where the sink is always placed in the middle of the simulation area. For the traffic, we considered usual CBR convergecast flows.

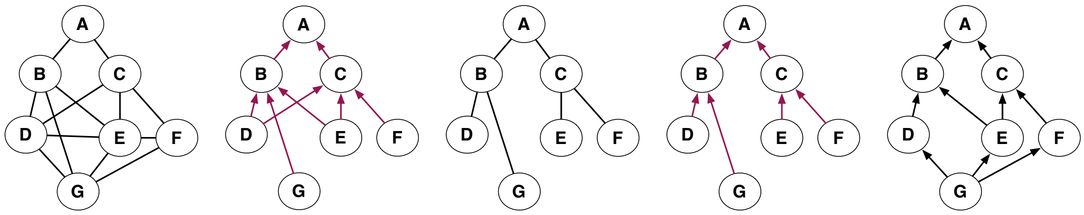
At the PHY layer, we used the path-loss shadowing model, calibrated with the scenario FB6 (indoor real deployment) presented in [CT11]: shadowing, path loss = 1.97, standard deviation = 2.0, $Pr(2m) = -61.4dBm$.

For the MAC layer, we used IEEE 802.15.4 in beacon mode with the extension of [PTD11], which implements the cluster-DAG topology. Let us explain this choice.

Even though theoretically, the MAC protocol does not create a routing topology, IEEE 802.15.4 behaves differently: it constructs a star, a peer-to-peer, or a cluster-tree. Since the network neighbors are a subset of the MAC neighbors, the topology obtained at the network layer depends heavily on the MAC layer.

When IEEE 802.15.4 constructs a peer-to-peer topology, RPL has several possible choices for the construction of the routing topology. For example, we can observe in Figure 5.4b the resulted DODAG constructed on top of the peer-to-peer topology from Figure 5.4a. Nodes like D and E are able to create redundant paths towards the sink. Hence, we decided to use this MAC topology in the previous chapter (with IEEE 802.15.4 beacon less).

However, it is practically impossible to use a peer-to-peer topology with the beacon mode of IEEE 802.15.4, since the nodes have to always keep their radio on. The cluster-tree topology is the only one suited for multi-hop networks that are energy constrained. Still, when IEEE 802.15.4 creates a cluster-tree topology, RPL does not have other path choices and must exploit a tree. If we take a look at Figure 5.4c, we can see a cluster-tree topology created by IEEE 802.15.4. RPL can choose only a subset of these links, and hence, constructs a DODAG that is identically to the IEEE 802.15.4 cluster tree (Figure 5.4d).



(a) IEEE 802.15.4 non-beacon: peer-to-peer (b) RPL on top of IEEE 802.15.4 non beacon (c) IEEE 802.15.4 beacon: cluster tree (d) RPL on top of IEEE 802.15.4 cluster tree (e) IEEE 802.15.4 beacon: cluster DAG

Figure 5.4: Topology control using hop count as a metric

Table 5.2: Simulation parameters

Parameter	Value
Simulation duration	7200 s
Number of nodes	50
Simulated area	300m x 300m, nodes placed randomly in a circle sink in the center
Traffic type, rate	CBR, 1 pkt/min
Data packet size	127 bytes (incl. MAC headers)
RPL	MinHopRankIncrease = 256
Trickle	$Imin = 2^7 ms$, $Imax = 16$, $k = 10$
MAC layer	802.15.4 mode beacon
MAC parameters	BO=7, SO =2
Energy consumption	CC2420 datasheet

In conclusion, we decided to use the Directed Acyclic Graph (DAG) structure proposed by Pavkovic *et al.* [PTD11]. As we can observe in Figure 5.4e, this topology manages to create redundant paths to the sink, and offers more choices for the construction of the routing DODAG.

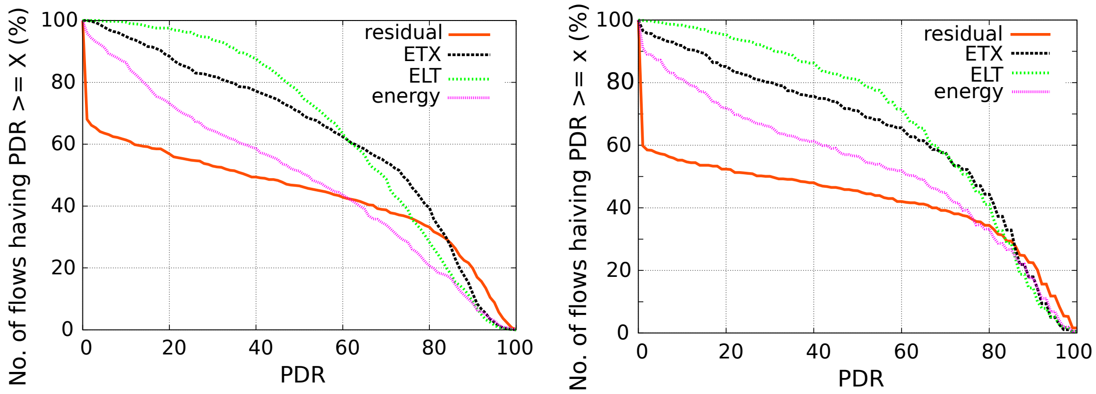
We configured RPL as illustrated in Table 5.2. As the simulated traffic is convergecast, `DAO` messages are deactivated. In order to evaluate our solution, we compared the ELT metric with the following routing metrics that take the energy consumption into account:

- the residual energy: the remaining energy of a node, as implemented by Kamgueu *et al.* [Kam+12];
- ETX (using the Minimum Rank with Hysteresis Objective Function [Gna12]): the average number of transmissions for an acknowledged packet;
- energy: a linear combination of ETX and the residual energy, as proposed by Chang *et al.* [Cha+13].

5.4.1 Reliability

We first evaluated the reliability of RPL as the end-to-end PDR for all the nodes. The PDR is computed as the ratio of the number of received packets by the border router, and transmitted by each node.

We can observe in Figure 5.5a the Complementary Cumulative Distribution Function (CCDF) of the end-to-end PDR for all the nodes. The residual energy has the worst end-to-end PDR, which does not come as a surprise, since it tends to privilege nodes with energy, without considering the quality of the links. A node ends up choosing bad radio links to forward its packets.



(a) Considering the whole period of simulation (2h) (b) After eliminating the first 1h of the simulation

Figure 5.5: Complementary CDF of the end-to-end PDR

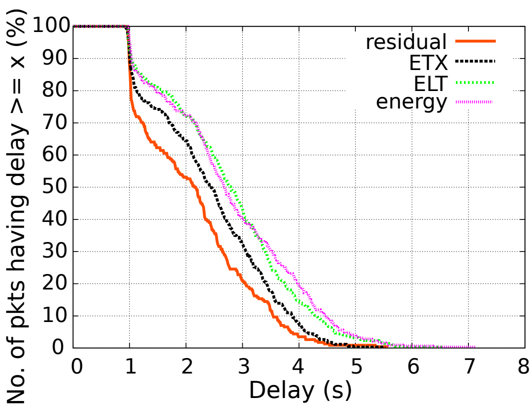


Figure 5.6: Complementary CDF of the end-to-end delay

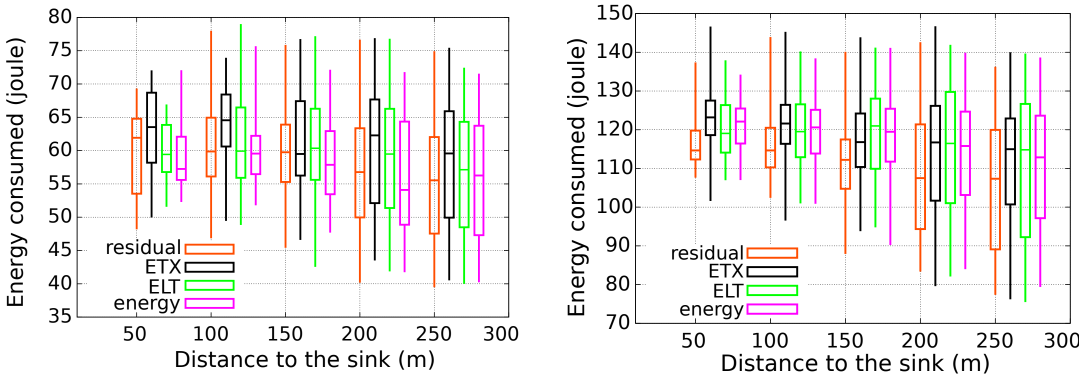
The energy metric accounts for the link quality, but its performance is degraded because of the linear combination with the residual energy. ELT on the other side, succeeds in having a PDR close to ETX. Moreover, the gap between the two metrics becomes smaller when we eliminate the first hour of the simulation, i.e., the bootstrap period (Figure 5.5b), ELT converges quickly to the best routes in energy while avoiding lossy links.

5.4.2 End-to-end Delay

Secondly, we evaluated the end-to-end delay, measured as the time between the packet generation at the source, and its reception by the border router.

We can see in Figure 5.6 that the difference between all the routing metrics is not very significant. The residual energy has the best delay because most of the packets are dropped, particularly by nodes far from the border router. Since the delay is computed only for received packets, nodes close to the sink (with shorter paths) are over-represented in the end-to-end delay result.

ELT and the energy metric have the worst delays. In the case of ELT, the delay is traded for the construction of more energy balanced paths. Indeed, nodes can decide to choose a less good link to balance the energy. Because of this, there will be more retransmissions at the MAC layer and hence, an increased end-to-end delay.



(a) After 1h of simulation

(b) After 2h of simulation

Figure 5.7: Energy consumption in function of the physical distance of the node from the sink

5.4.3 Energy Consumption

We also evaluated the energy consumed by the nodes after 1 hour of simulation, and respectively, after 2h hours. Figure 5.7 presents the box plot for the energy consumption of the nodes in function of their physical distance to the sink.

Let us first take a look at the energy consumed during the bootstrap period (i.e., after 1 hour of simulation). We can see in Figure 5.7a that the residual energy and the energy metric have less energy consumption on average. However, we have to not forget that their PDR is considerably lower than when ETX and ELT are used. The less packets a node will transmit, the less energy it will consume.

ELT has a better energy consumption than ETX. Moreover, ELT manages to balance the energy consumption over all the nodes, having an average consumption of 60 joules. As the simulation continues, ELT still keeps an energy-balanced routing structure (Figure 5.7b). We can also notice that the worst case scenario is the same, regardless the distance from the border router. This shows that ELT achieves our objective of energy balancing.

5.4.4 Lifetime

We measured the lifetime of the network (i.e., the time until the first node will run out of energy) using Equation 5.5, in function of the density. We kept the same simulation area and we increased the number of nodes. We can observe in Figure 5.8 that ELT manages to double the network lifetime when compared to ETX.

The lifetime decreases when the network becomes more dense, since the bottlenecks will have to relay more packets. Still, ELT outperforms all the other routing metrics. ETX manages to have better lifetime than the residual energy because it chooses good quality links that do not require many retransmissions.

We can see a significant difference between the ELT when the network has 50 nodes and 70 nodes. This is due to the fact that the MAC layer has difficulties managing the traffic when the network becomes denser: a lot of packets are dropped because too much time spent in the buffer. The corresponding links will be evaluated by the ETX as poor, and hence the ELT will decrease.

5.4.5 Network Dynamics

Finally, we studied the dynamics of the routing topology by calculating the number of times each node changes its preferred parent. Figure 5.9a illustrates the CCDF of the number of parent changes, during the whole simulation period. We can observe that all the metrics but the residual energy, show an increased number of parent changes. This is due to the fact that they

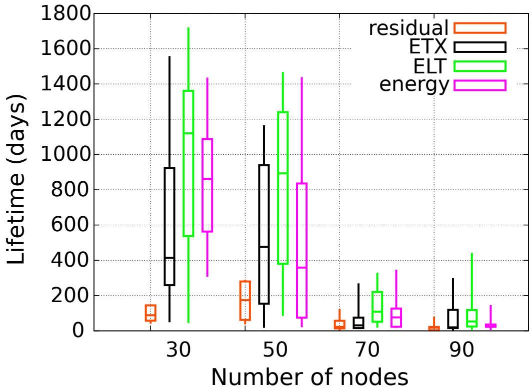
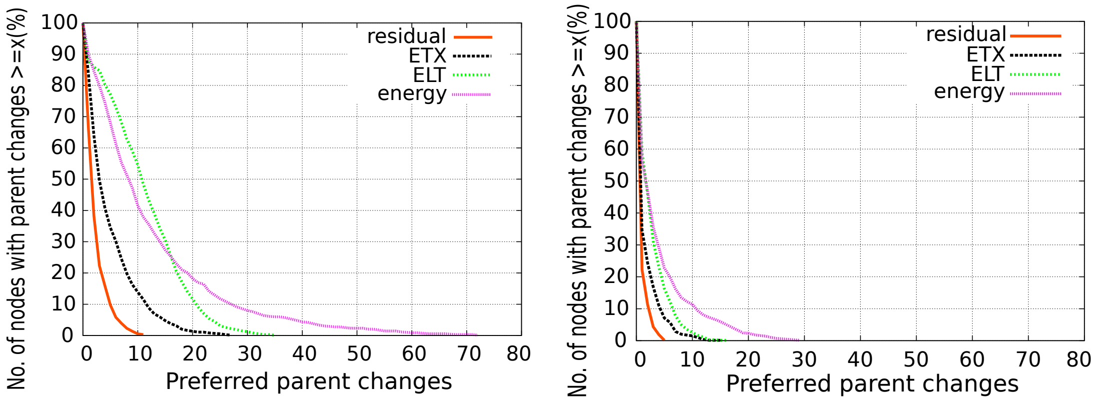


Figure 5.8: Network lifetime (time until the first node will run out of energy) in function of the no. of nodes



(a) Considering the whole period of simulation (2h) (b) After eliminating the first 1h of the simulation

Figure 5.9: Complementary CDF of the number of parent changes

use ETX for the estimation of the link quality, which we have shown that induces instability in the network.

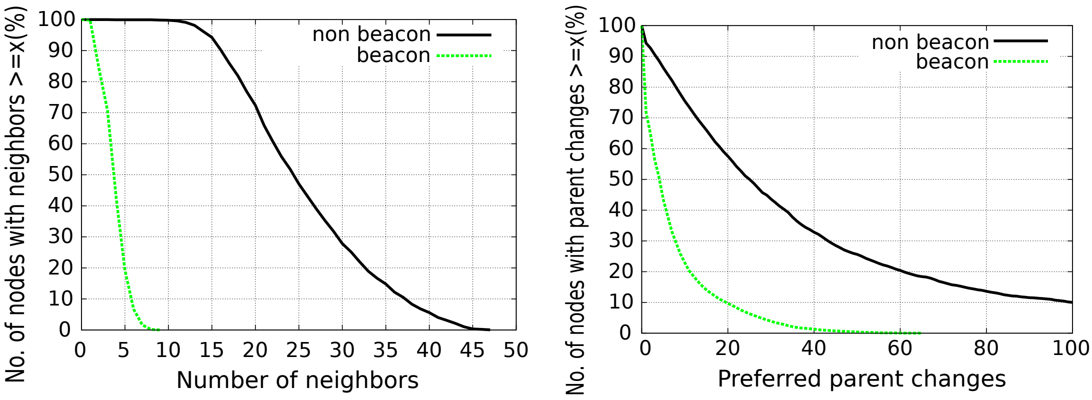
Even after we eliminated the bootstrap period, the energy metric continues to induce a significant overhead, while all the other metrics considerably decrease their number of parent changes (Figure 5.9b). This explains why it is the only metric that does not decrease its average energy consumption once the simulation has passed its period of bootstrap period (Figure 5.7).

As a general remark, we can notice that overall, the network is quite stable when compared to the results that we have obtained in the previous chapter. This difference is due to the protocol that we used at the MAC layer: IEEE 802.15.4 in beacon mode (with the cluster-DAG topology) vs. IEEE 802.15.4 beacon less (with the peer-to-peer topology).

Indeed, each topology has a different impact on the dynamics of the network. The peer-to-peer topology is more redundant and hence, RPL may have the choice between more parents than when the cluster-DAG is used. Indeed, if we take a look at Figure 5.10a, we can see that a node has on average 5 times more neighbors when the DODAG is constructed on top of the beacon less peer-to-peer topology, than on top of the cluster-DAG with the beacon mode.

It is only natural that the number of neighbors influences the number of preferred parent changes. The more neighbors a node has, the more parents with close ranks it has. Figure 5.10b shows that the number of preferred parent changes can increase up to 3 times when the peer-to-peer topology with the non beacon mode is used.

However, reducing the number of neighbors (i.e., routing possibilities) should not be used as



(a) CCDF of the number of neighbors

(b) CCDF of the number of parent changes

Figure 5.10: Impact of the MAC topology when using ETX as a routing metric

a solution for minimizing the number of preferred parent changes: it also imposes a limit on the diversity and robustness of the routes.

5.5 Conclusion

We highlighted that none of the routing metrics proposed to be used with RPL focuses on globally improving the network lifetime. In consequence, we designed a new routing metric to prolong the network lifetime: the *Expected Lifetime* (ELT). ELT estimates the time at which a node will run out of energy in the current traffic conditions, according to its residual energy and the link quality to its preferred parent.

We also showed how to efficiently implement ELT with RPL, by proposing a new algorithm for preferred parent selection. A node constructs energy balanced paths by considering not only the lifetime of the bottleneck, but also its own lifetime. As a result, ELT manages to maximize the lifetime of the network.

In our approach, we considered that the energy consumed to receive a packet can be neglected. We could further optimize the ELT metric by taking into accounting the reception energy consumption. For this, a node needs to know the ETX of the links from its children to itself, which is difficult to obtain without increasing the overhead in the network.

To evaluate our proposal, we compared ELT with three other metrics: ETX, the residual energy, and a linear combination of ETX and the residual energy. The ELT metric manages to have performance results close to ETX in terms of reliability and delay, while having less energy consumption. Indeed, RPL constructs a more energy balanced topology, which leads to an increase in network lifetime when compared to the other metrics.

Still, ELT uses ETX in its computation, and it is by design prone to induce instability in the network. A small variation in the link quality estimation is sufficient to trigger the change of the next hop. To smoothen the values of ETX, we have used the Exponential Weighted Moving Average (EWMA) estimator. However, the stability could be improved by using a more accurate estimation technique.

To counteract this drawback, we propose in the next chapter a multiparent version of RPL.

Chapter 6

Multiparent Routing with RPL

To improve the network lifetime, we have proposed in the previous chapter the *Expected Lifetime* routing metric and showed how to implement it using RPL. Even though simulation results showed that we manage to maximize the network lifetime, we think there is still room for improvement.

With RPL, a node selects one preferred parent to construct the Destination-Oriented Directed Acyclic Graph (DODAG) without loops, and to compute its own rank. However, only this preferred parent is used for routing: the other ones have just a *backup* purpose. We are convinced that we should rather exploit this diversity to distribute the traffic load in the network, so that we improve the network lifetime.

Hence, we propose to enhance RPL by exploiting several parents, in order to create energy-balanced paths. We identify the weakest nodes in the network (in terms of energy) by using the *Expected Lifetime* metric, and we construct accordingly the DODAG. Then, we probabilistically forward the traffic through all the parents, so that each path consumes the same quantity of energy.

By using a multiparent approach we also address the stability problem emphasized in Chapter 4. A node will change its preferred parent only when it becomes useless i.e., it does not forward traffic anymore. Hence, the number of DODAG reconfigurations are reduced, which minimizes the energy consumption.

Contribution

This chapter presents the following contributions:

1. We propose a multiparent version of RPL, and we combine this approach with the *Expected Lifetime* routing metric to further maximize the network lifetime;
2. We present a load-balancing algorithm to split the traffic among several paths, while balancing the energy equally among them;
3. We propose an algorithm to smoothen path metric variations, in order to avoid sudden traffic redirections, reducing the route changes;
4. We evaluate the multiparent version of RPL, showing the gains when compared to the standard version.

6.1 Problem Statement

RPL constructs in a distributed mode a Destination-Oriented Directed Acyclic Graph (DODAG) rooted at the border router. However, all the traffic is sent to the preferred parent: the other parents are used just for backup purposes.

RPL specifies that the preferred parent of a node may be, conceptually, *a set of multiple parents if those parents are equally preferred and have identical rank* [Win+12]. However, the standard does not specify the forwarding rule, and to the best of our knowledge, all the implementations consider the preferred parent as one single node.

With this in mind, the current version of RPL faces the following drawbacks:

1. **Even though RPL creates a DODAG, for routing it uses a tree topology, which does not allow for load balancing.** Indeed, a node has to take a binary decision: each of its parents receives either all its traffic or none. *Forwarding the traffic to several parents would result in a more reliable and energy efficient routing protocol.*

2. **Estimating the quality of a radio link is energy consuming.** In order to save energy, a passive measurement technique is preferable. However, only the link quality to the neighbors with which a node exchanges data packets can be estimated passively. Since in RPL all the traffic is sent to the preferred parent, a node may only estimate the link quality to this node. However, *by balancing the traffic to all the parents, a node could continuously estimate the quality of all these links.*

3. **Frequent preferred parent changes can induce instability in the network and are energy consuming.** Indeed, the change of the preferred parent triggers the reset of the trickle timer, which causes a more frequent `DIO` transmission. The more control packets are sent, the more energy is consumed by the nodes. Moreover, frequent changes can induce instability in the network, which are exacerbated with larger topologies [ITN13]. *By using all the parents to route the traffic, the preferred parent could be changed only when it is not useful anymore, reducing the trickle timer resets and the probability of creating instabilities.*

6.2 Multiparent Routing for Network Lifetime Maximization

In order to maximize the network lifetime, we will first generalize the ELT metric to the multiparent routing. In particular, we have to consider the fact that a node will send its data packets to several parents with different link qualities. Hence, the energy consumed by the node will depend on:

- the amount of traffic sent to each parent. Let α_P be the ratio of traffic sent to parent P;
- the link reliability to each of its parents, i.e., $ETX(N, P)$.

The average number of MAC transmissions can then be computed as:

$$\sum_{P \in Parents(N)} \alpha_P \times ETX(N, P), \text{ where } \sum_{P \in Parents(N)} \alpha_P = 1 \tag{6.1}$$

The computation of ELT from Equation 5.5 becomes in the multipath scenario (using the notation from Table 6.1):

$$ELT(N) = \frac{E_{res}(N)}{T_N \times \frac{\sum\limits_{P \in Parents(N)} \alpha_P \times ETX(N,P)}{\text{DATA\_RATE}} \times P_{TX}(N)} \tag{6.2}$$

Second, ELT identifies the bottleneck of a path, and forwards the traffic to the parent that maximizes the network lifetime. In a multiparent scenario, a node has to deal with multiple bottlenecks, from all its parents. Several challenges arise. We must:

Table 6.1: Notation used for the ELT metric in the multiparent scenario

Notation	Meaning
ELT(X)	Expected lifetime of X
$\boldsymbol{E_{res}(X)}$	Residual energy of X (in Joule)
$\boldsymbol{P_{TX}(X)}$	Radio power in transmission mode (in Watt or Joule/s)
ETX(A,B)	ETX of the link $A \to B$
$\boldsymbol{T_X}$	Throughput (bits/s) of X
$\boldsymbol{\alpha_P}$	Ratio of traffic sent to parent P
$\boldsymbol{\delta}$	used for the computation of α_P
$\boldsymbol{r_{X,B}}$	Ratio of traffic forwarded by X to bottleneck B
$\boldsymbol{T_{gen}(X)}$	Traffic generated by X
Children(X)	Children set of node X
Parents(X)	Parents set of node X
Bottlenecks(X)	Bottlenecks set of node X
DATA_RATE	The rate at which the data is sent (bits/s); All nodes transmit at the same rate

1. find the nodes that will be the first ones to run out of energy (i.e., the bottlenecks) on each path, and advertise them in a compact manner;
2. construct an energy-balanced topology using multiple parents;
3. balance the traffic to all the parents, while taking into account the lifetime of each bottleneck.

We will now address each of these challenges in the following sections.

6.3 Multiple Bottlenecks: ELT Estimation and Advertisement

In the multipath scenario, a node will send its traffic through several paths, i.e., it will have to maintain information about several bottlenecks. Moreover, only a part of its traffic will arrive at a specific bottleneck. We present here how to compute the ELT of a bottleneck in the multiparent scenario, and how to send the information about the bottlenecks along the path in a compact manner.

6.3.1 ELT Estimation of Bottleneck

When a node N has to associate with the DODAG, it needs to estimate the impact of its traffic on the lifetime of a bottleneck B. In the case of one path, it was sufficient for N to add its traffic (T_N) to the throughput of the bottleneck:

$$\mathbf{new\_T_B} = T_N + T_B \tag{6.3}$$

and then estimate the lifetime of B as:

$$ELT(B) = \frac{E_{res}(B)}{\mathbf{new\_T_B} \times \frac{\sum_{P \in Parents(B)} \times \alpha_P \times ETX(B,P) \times P_{TX}(B)}{\text{DATA\_RATE}}} \tag{6.4}$$

However, in a multipath scenario, we have to consider that a node sends its traffic to several parents. Hence, only a part of its traffic will finally arrive at a specific bottleneck. We need to determine the ratio of traffic that a node forwards to a bottleneck.

Let $r_{N,B}$ be the ratio of traffic that N forwards to the bottleneck B. Given a parent P, N computes the ratio of traffic that will reach B through P as the product of the proportion of

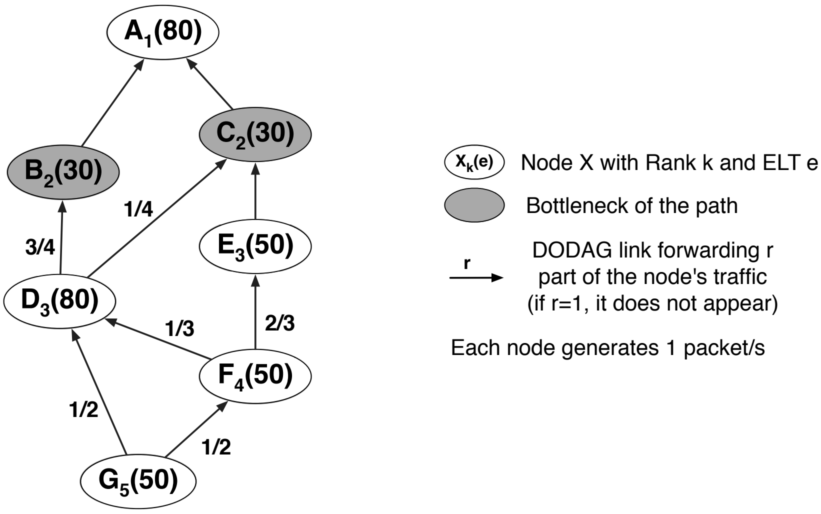


Figure 6.1: Path selection with ELT (`MinHopRankIncrease`=1)

traffic that it sends to P (α_P) and the ratio of traffic that P forwards to the bottleneck B ($r_{P,B}$). Then, it will sum over all its parents these values. More formally:

$$r_{N,B} = \sum_{P \in Parents(N)} (\alpha_P \times r_{P,B}) \tag{6.5}$$

In the case when a node is itself the bottleneck, the ratio of the traffic that it forwards to the bottleneck is equal to 1 (i.e., $r_{B,B} = 1$).

Let us take an example. Consider node G and its parents D and F in Figure 6.1. 3/4 of the traffic of D is forwarded through the bottleneck B. This is also the case of $1/3 \times 3/4$ of the packets of F. Finally, the actual quantity of traffic of G that reaches B is: $1/2 \times 3/4 + 1/2 \times 1/3 \times 3/4$.

Thus, every node computes recursively the ratio of traffic forwarded to a bottleneck by summing over all the parents the ratio of traffic forwarded to it. Consequently, the new throughput of the bottleneck B can be estimated as:

$$\mathbf{new\_T_B} = \mathbf{r_{N,B}} \times T_N + T_B \tag{6.6}$$

Knowing this information, a node N can now estimate its impact on the lifetime of B using Equation 6.4 with the newly computed throughput of B from Equation 6.6.

6.3.2 Compact `DIO` Advertisement

A node maintains the list of all its bottlenecks. This information has to be updated and included in each of its `DIO`. Consequently, we are able to determine partially overlapping paths: several parents may lead to the same bottleneck.

Besides the information advertised in the one path scenario (*id*, *existing_ traffic*, and *B_ const*), a node also has to advertise the ratio of traffic that itself forwards to the bottleneck ($r_{N,B}$). To sum up, a `DIO` will contain a list of bottlenecks with the following information for each of them:

- **id:** the bottleneck id: IPv6 address (16 bytes), 6LowPAN address (4 bytes) or IEEE 802.15.4 short address (2 bytes);
- **ratio:** the normalized value of the ratio of traffic forwarded by the node to this bottleneck, which is computed recursively by each node: $r_{N,B}$ (1 byte);

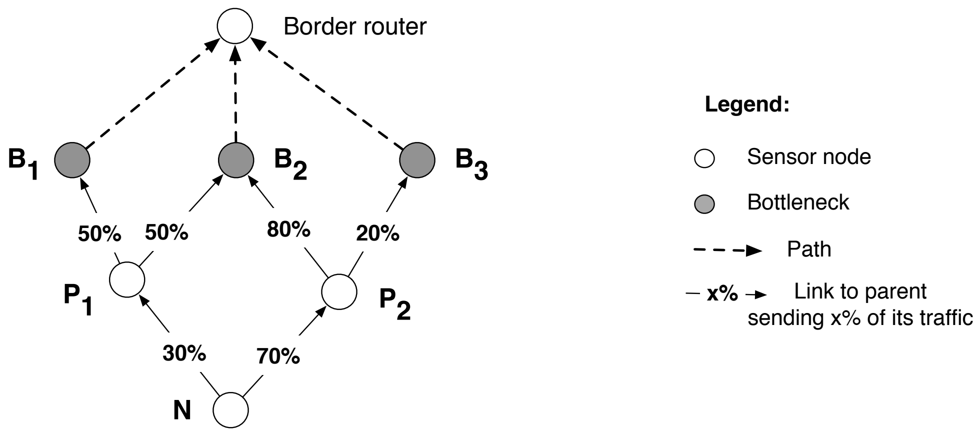


Figure 6.2: Number of bottlenecks

- **existing_traffic:** the traffic forwarded by the bottleneck to the sink (normalized): T_B (1 byte);
- **B_const:** the normalized value of the bottleneck constant (2 bytes) computed in the multiparent scenario as:

$$B\_const = \frac{E_{res}(B)}{\frac{\sum_{P \in Parents(B)} \alpha_P \times ETX(B,P) \times P_{TX}(B)}{\text{DATA\_RATE}}} \tag{6.7}$$

Practically, a tradeoff exists between the accuracy of the information and the overhead induced. On one side, if a node advertises too few bottlenecks, the lack of information could lead to less energy balanced paths. For example, in Figure 6.2, if $P1$ only advertises $B1$ as a bottleneck, and $P2$ only $B3$, N could end up significantly reducing the lifetime of the bottleneck $B2$. On the other side, if the number of bottlenecks advertised is very large, it can induce more overhead in the network, and make the nodes consume more energy.

We will study this tradeoff in the performance evaluation section. We verified that advertising a limited number of bottlenecks is sufficient to practically balance the energy consumption in the network. Still, let us find the superior bound.

Worst case analysis

Let us now analyze the maximum number of bottlenecks that a node N can have. This will happen when all the bottlenecks are independent from each other, i.e., they are not shared by one or more nodes. The maximum number of bottlenecks will be reached when:

- all the bottlenecks are situated at one-hop from the border router and are distinct from each other (i.e., the furthest away possible from the nodes);
- all the nodes (but the bottlenecks) have the maximum number of parents k, and all the parents are distinct from each other. This condition makes a node advertise all the bottlenecks from all the parents as its own bottlenecks.

If $k = 3$, the topology created following these conditions corresponds to what we can see in Figure 6.3a. Now, if we eliminate the border router from the figure, we can notice that we have a perfect k-ary tree (Figure 6.3b, with $k = 3$). The problem of finding the maximum number of bottlenecks is equivalent to computing the total number of leaves in this tree. If k represents the

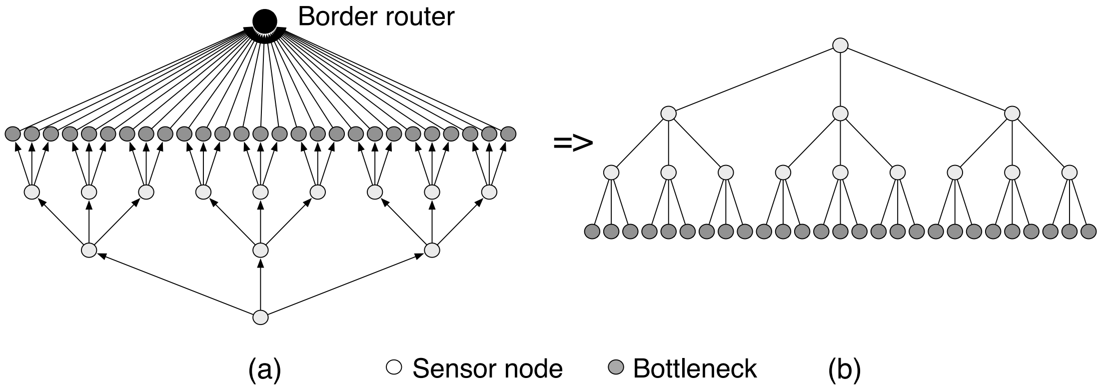


Figure 6.3: Maximum number of bottlenecks

number of children of a node and *depth* the depth of the tree, then the total number of leaves is equal to k^{depth}. If we go back to our topology (i.e., we add another level when we put back the border router), the total number of bottlenecks will be $k^{depth-1}$.

In conclusion, the maximum number of bottlenecks that a node N can have is the maximum number of parents that a node has in the network at the power *(depth of* $N-1$*)*:

$$\left(\max_{M \in Network} |Parents(M)|\right)^{depth_N - 1}$$

However, since in most cases, the bottlenecks will be shared among one or more nodes, this scenario will rarely occur in real-life topologies.

6.4 Constructing an Energy-Balanced DODAG

We now define how a node may choose its preferred parent, and how to construct a loop-free topology using the ELT metric, in the multiparent scenario.

6.4.1 Preferred Parent Selection

When choosing its preferred parent, a node must consider both its own lifetime and the lifetime of the bottlenecks, in order to estimate which of them becomes the new bottleneck. However, it is not possible to know the ratio of traffic that will be sent to each of the parents before actually choosing them. Hence, it is challenging to accurately estimate both the lifetime of the node, and the lifetime of the bottlenecks.

In order to balance more efficiently the energy consumption, we consider the worst case, i.e., a node sends all its traffic to a single parent. In consequence, during the preferred parent selection we underestimate the lifetime of the most constrained bottlenecks. In this way, we choose as preferred parent the node maximizing the lifetime of all the bottlenecks.

We consequently propose Algorithm 2 (a modification of Algorithm 1 from the one path scenario) to select the preferred parent (using the notation of Table 6.1). For each possible parent (i.e., a neighbor advertising a rank smaller than itself) a node N:

1. computes the ELT of all the bottlenecks advertised by a parent P, as if it will send all its traffic to that parent and save the minimum ELT value among all of them (line 6);

2. computes its own lifetime when choosing this parent and verifies if N becomes the new bottleneck (line 7);

Algorithm 2: Preferred parent selection

```
   Data: N
   Result: preferred_parent of N
 1 max_elt ← 0;
 2 for P ∈ Parents(N) do
 3     // all the traffic is sent to P
 4     α_P ← 1;

 5     // track the minimum ELT (all bottlenecks & myself)
 6     min_elt ← min_{B∈Bottlenecks(P)} {ELT(B)};
 7     min_elt ← min{min_elt, ELT(N)};

 8     // is this parent the best one?
 9     if max_elt < min_elt then
10         max_elt ← min_elt;
11         preferred_parent ← P;
12     end

13     // test now the other parents
14     α_P ← 0;
15 end
16 return preferred_parent;
```

3. removes the traffic to this parent to test the other ones: it will iteratively test all the parents before making a decision (line 14);

4. chooses as preferred parent the node that maximizes the lifetime of the bottleneck with the minimum ELT, itself included (lines 9, 10, 11).

During the bootstrap of the network, a node does not know the link quality to its neighbors. To avoid choosing as preferred parent the sender of the first received `DIO`, a node waits a period of time, to be able to receive several `DIO` before making a decision. During this period, the node adds in its parents set all the `DIO` senders. After a predefined time, it triggers the selection of the preferred parent using the algorithm presented above.

6.4.2 Path Maintenance and Discovery of Better Bottlenecks

To fully exploit the flexibility of RPL, we propose to separate the metric that we use to construct the DODAG from the metric used to compute the rank. Indeed, we use ELT to avoid the most constrained nodes in the network, and ETX to ensure loop-freeness. Since RPL forbids a node to consider as next hop a neighbor with higher rank than itself [Win+12], we construct from the beginning a loop-free topology. In order to keep the loop-freeness, a node:

1. chooses its preferred parent using Algorithm 2;

2. computes its own rank based on the rank of its preferred parent (from Equation 5.10);

$$\begin{aligned} Rank(N) &= Rank(P) + Rank\_increase \\ Rank\_increase &= ETX(N,P) \times \texttt{MinHopRankIncrease} \end{aligned} \tag{6.8}$$

   where P is the preferred parent of N and `MinHopRankIncrease` the RPL parameter [Win+12].

3. removes from the parents set all the neighbors having a rank higher than itself;

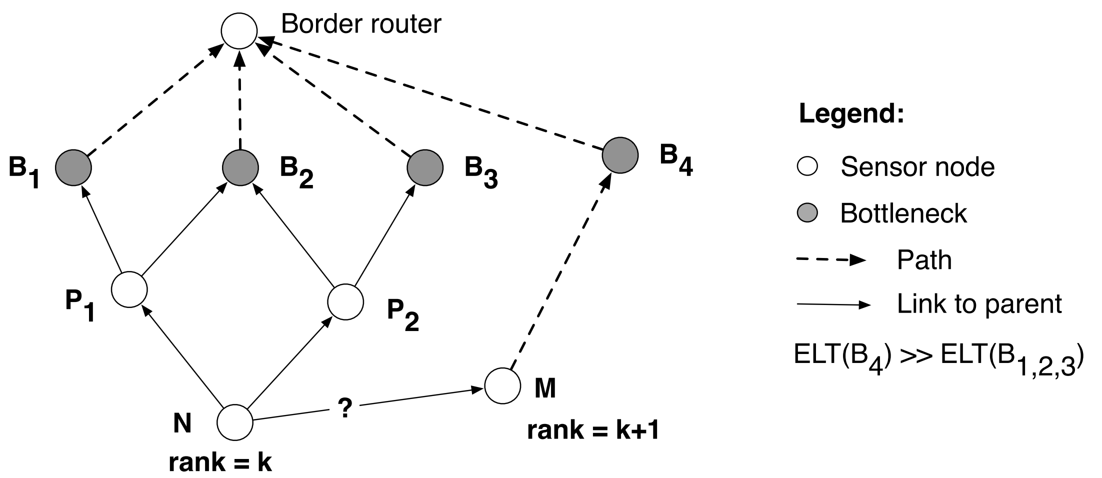


Figure 6.4: Discovering new paths

4. aggregates the bottlenecks and updates the corresponding information in its `DIO`;
5. ignores all `DIO` from nodes advertising a higher rank than itself, in order to avoid the creation of loops.

One important question arises: what if there exists a path advertising a higher ELT, but the node will not receive this information, since the neighbor advertising it has a higher rank? For example, in Figure 6.4, the node M advertises the bottleneck B_4 with a lifetime greater than all the other bottlenecks. However, since $Rank(M) > Rank(N)$, N will never find this path, even though M is not situated in its sub-DODAG.

We propose here to allow a node to consider a `DIO` advertising a larger rank than itself, under the condition of maintaining a loop-free topology. However, we have to introduce some conditions to forbid such attachment if M is in the sub-DODAG of N.

Let us assume that M advertises a bottleneck B. If B is not one of the bottlenecks of N, this means the sub-DODAGs are at least partially disjoint: another path exist through the bottleneck B.

Hence, a node N considers M with $Rank(M) \geq Rank(N)$ as preferred parent if it satisfies the following conditions:

1. the bottlenecks advertised by M are not all also bottlenecks of N:

$$Bottlenecks(M) \not\subseteq Bottlenecks(N)$$

2. M advertises at least one bottleneck $B_{new} \in Bottlenecks(M)$ whose lifetime is greater than the maximum ELT of all the bottlenecks of N:

$$ELT(B_{new}) > \max_{B \in Bottlenecks(N)} \{ELT(B)\} \tag{6.9}$$

Moreover, we need to make sure that the ELT of B_{new} will not become smaller than the maximum ELT of all the bottlenecks of N, once the traffic of N will be sent to M. Equation 6.9 becomes:

$$ELT(B_{new}) > \max_{B \in Bottlenecks(N)} \{ELT(B) | r_{N,B} = 0\} \tag{6.10}$$

where $r_{N,B}$ represents the ratio of traffic forwarded by N to the bottleneck B (cf. Equation 6.5).

Algorithm 3: Load balancing

```
   Data: N, δ - the step of the load increase
   Result: compute {α_P}_{P∈Parents(N)} — the ratio of traffic to send to each parent;
1  for i = 1 to δ^{-1} do
2      max_elt ← 0;
3      for P ∈ Parents(N) do
4          // test this parent P with its new weight
5          α_P ← α_P + δ;
6          // track the min ELT with this new weight
7          min_elt ← min_{B∈Bottlenecks(P)} {ELT(B)};
8          min_elt ← min{min_elt, ELT(N)};
9          // is this parent the best one?
10         if max_elt < min_elt then
11             max_elt ← min_elt;
12             parent_max ← P;
13         end
14         // test each parent before taking a decision
15         α_P ← α_P − δ;
16     end
17     α_parent_max ← α_parent_max + δ;
18 end
```

These conditions allow a node to consider a `DIO` from a neighbor situated deeper in the DODAG, and hence, find better paths, while maintaining the loop freeness of the topology.

6.5 Load Balancing for Lifetime Maximization

After constructing the DODAG with useful multiparents, we now have to address the problem of the forwarding plane. A node must split its traffic among all the available parents, while considering the lifetime of each bottleneck. We have to propose a heuristic to determine the weights associated to each parent so that all the paths have the same lifetime.

The first solution would consist in modeling the problem with linear inequalities:

- the weights of all parents represent the set of unknown variables;
- the node must compute the lifetime of each bottleneck, considering its own traffic;
- the optimal solution would consist in maximizing the minimal lifetime of all the bottlenecks.

This linear formulation may also be injected into a linear solver [Kad+05]. However, we consider the extra memory and CPU consumed by this solution inadequate for small sensor nodes.

Hence, we present here a greedy heuristic. A node N has to distribute the load to each parent so that it balances the expected lifetime of the corresponding bottlenecks. Consequently, a node divides its traffic in $\frac{1}{\delta}$ equal fractions, and assigns sequentially each fraction to the parent which maximizes the minimum lifetime among all its bottlenecks.

Algorithm 3 defines the heuristic more formally:

1. First, N tries to find the best parent to send δ of its traffic by iteratively testing each parent (line 3):

(a) N computes the minimum ELT that would be obtained by increasing the weight of this parent by δ (line 5). It considers the lifetime of each bottleneck (line 7) and of its own (line 8);

(b) If this minimum value maximizes the network lifetime, it saves the current parent as the best one (line 10-13);

(c) N removes the hypothetical δ from the ratio of traffic to be sent to this parent (α_P), in order to test the other parents before definitively setting the new weight (line 15);

2. Finally, N assigns δ of the total traffic to the best parent (line 17) and re-starts with the next δ (line 1).

It is obvious that the smaller δ is, the closer to the optimal the solution will be. To help accomplish this, we put the condition for δ to be smaller than the inverse of the maximum number of parents. Indeed, if we reconsider the example from Figure 6.2. N has two parents: P_1 and P_2. The traffic load associated to each of the parents is $\frac{1}{2}$. If δ would be greater than the inverse of the maximum number of parents (i.e., $\delta > \frac{1}{2}$), an optimal repartition of the weights would simply not be possible.

6.5.1 Complexity

For each parent, the algorithm searches the bottleneck having the smallest ELT. Since the minimum value in a list can be found in $O(n)$ and this value must be searched for each parent, the complexity of the greedy algorithm is $O(n * n) = O(n^2)$.

Some optimizations are possible in the implementation. In particular, for $i > 1$, a node has to recompute the minimum ELT (lines 5-15) only for the parent which was the best one at the previous iteration ($i - 1$). Indeed, the possible weight of all the other parents has already been considered in the previous iteration.

6.5.2 Correctness: $(1 + \delta)$ - approximation

As previously said, the smallest δ is, the closest to the optimal the solution will be. However, there might be cases where the optimal values for the parent weights are not found.

Let us consider still the example in Figure 6.2. P_2 has to send $\frac{4}{5}$ of its traffic to B_2 and $\frac{1}{5}$ to B_3. Let us take $\delta = \frac{1}{2}$, the maximum value allowed (we put the condition that δ should be smaller than the inverse of the maximum number of parents). Algorithm 3 will first allocate $\frac{1}{2}$ of its traffic to parent B_2. Then, the optimal choice would be to allocate another $\frac{3}{10}$ of its traffic to B_2 and the rest of $\frac{1}{5}$ to B_3. However, it will allocate the last δ of its traffic ($\frac{1}{2}$) to B_2, hence, being sub-obtimal. Indeed, 100% of the traffic will be forwarded to B_2 and 0% to B_3.

Theorem 6.1. *The presented greedy algorithm is a $1 + \delta$ approximation.*

Let us give an intuitive explanation. The full proof can be found in the Appendix A.1.

We first take the particular case when a single node in the network (N) has packets to send. The greedy algorithm makes the choice that seems best at each step, i.e., when each δ is distributed. Since δ is a constant, it means that the algorithm might have a problem distributing the last δ of the traffic.

Let us assume that the last δ of the traffic is sent by N to the parent P_1, which is not optimal. If we focus on the most constrained bottleneck (e.g., B_1), we can distinguish two cases:

1. P_1 is the only parent of N forwarding traffic to the bottleneck B_1 (Figure 6.5a). B_1 will be overloaded with all the extra traffic from N. Hence the algorithm is a $(1 + \delta)$ - approximation.

2. All the traffic from P_1 plus some of the traffic from other parents (e.g., P_2) is forwarded to the bottleneck B_1 (Figure 6.5b). This means that B_1 will be overloaded with the traffic sent from P_1. At the same time, less traffic than the optimal is sent on the other parents, and so, B_1 will be less loaded with that corresponding traffic.

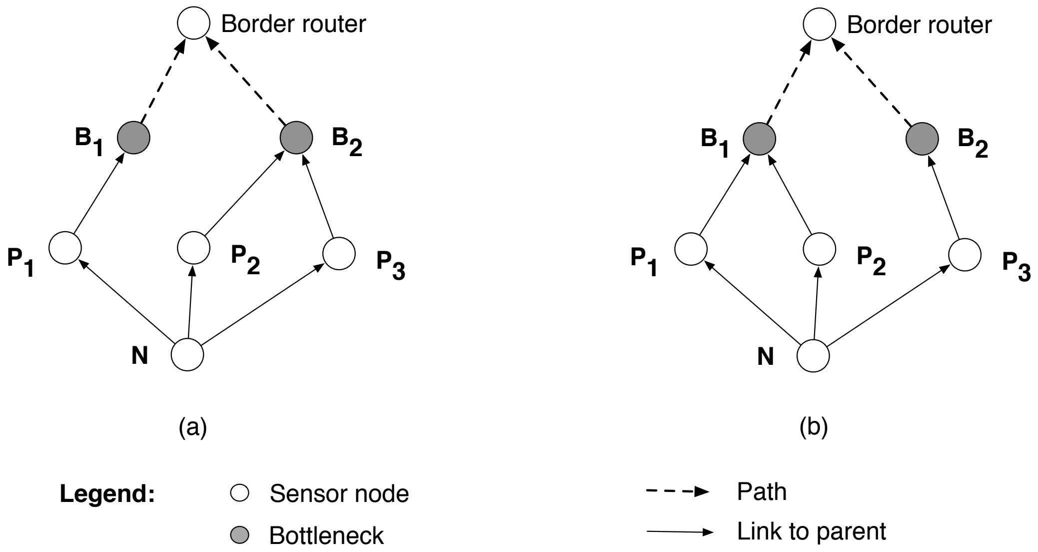


Figure 6.5: Correctness

In conclusion, the bottleneck B_1 will be overloaded with at most δ of the traffic of N, hence the algorithm is a $(1 + \delta)$ - approximation.

The reasoning holds also for when there is more than just one node transmitting in the network. If each node is at $(1 + \delta)$ from the optimal, globally it will also be at $(1 + \delta)$ from the optimal.

6.5.3 Maintaining the Stability

Frequent preferred parent changes can induce instability in the network. Some would argue that these instabilities are beneficial since it shows that the network adapts to changes. However, DODAG reconfigurations have a strong impact on the performance of the network, especially on the end-to-end packet delivery ratio and on the energy consumption [ITN13].

We address this problem as follows:

1. we change the preferred parent only when it is no longer useful;
2. we gradually redirect the traffic from one parent to the others when recomputing the loads.

Changing the preferred parent

When a node changes its preferred parent, it triggers the reset of the trickle timer. This means that the node sends control packets more frequently and hence, consumes more energy. Besides, parent changes also imply traffic redirection. Since the metric depends on the traffic forwarded by the bottlenecks, all the nodes must in this case update their path metric.

We change the conventional meaning of the preferred parent (i.e., the parent offering the best path to the border router) to a parent that a node uses to compute its rank, without necessarily being the best one. We aim here at reducing the number of preferred parent changes while keeping up-to-date information about all the parents.

We propose to adopt a conservative approach in the maintenance of the parent list:

1. a node removes a parent P only if this parent is not useful anymore, i.e., the traffic that is actually forwarded to P (α_P) is smaller than a threshold value. If P was the preferred parent, the node re-executes the Algorithm 2 to select the new preferred parent and updates accordingly its rank;

Algorithm 4: Parent weight normalization

Data: N, α_{max} – the maximum increase/decrease in the parent weight,
$\{old\_\alpha_P\}_{P \in Parents(N)}$ – the current ratio of traffic for each parent,
$\{new\_\alpha_P\}_{P \in Parents(N)}$ – the newly computed ratio of traffic for each parent;

Result: $\{\alpha_P\}_{P \in Parents(N)}$ – the new ratio of traffic, after normalization;

1 **for** $P \in Parents(N)$ **do**
2   `// compute the difference between the old weight and the new one`
3     $diff_P \leftarrow old\_\alpha_P - new\_\alpha_P$;
4 **end**
5 **for** $P \in Parents(N)$ **do**
6   `// limit the weight change to` α_{max} `and normalize the values`
7     $\alpha_P \leftarrow old\_\alpha_P + diff_P \times \alpha_{max} \times \frac{1}{\max\limits_{P \in Parents(N)} diff_P}$;
8 **end**

2. any neighbor with a lower rank is inserted in the parent list. The node N dynamically updates the weight of all its parents (α_i, $i \in Parents(N)$) every time a new `DIO` is received.

Redirecting the traffic

A small variation in the link quality estimation is sufficient to change the preferred parent (local maximum among the neighbors). As the inaccurate estimation of the radio link quality has a significant impact on the stability of RPL [ITN13], we need to also take into account small and transient inaccuracies.

Since the link quality influences the quantity of traffic on a path, and vice versa, minimizing the variations of the link quality estimation is equivalent to minimizing the variations of the throughput.

Consequently, we propose here to progressively redirect the traffic when recomputing the weights for each parent, instead of completely replacing the old values with the newly computed ones. We set a maximum value for the increase/decrease of the parent weight between two estimations. We let the network adapt progressively to this change, while avoiding sudden redirections due to metric variations.

We propose Algorithm 4 for re-computing the parent weights:

1. For each parent P, a node N computes the difference between the old weight and the new one (line 3);

2. N adds to the old weight (old_α_P) the normalized value of the difference computed in the previous step ($diff_P$) (line 7).

This algorithm gradually redirects the traffic, reducing thus, useless oscillations in the network and saving energy.

6.6 Performance Evaluation

For the performance evaluation, we used the WSNet simulator with the same hypothesis as in the previous chapter. We configured RPL as illustrated in Table 6.2 and we compared the following variants:

- a standard version of RPL: only the preferred parent is used to forward packets;

Table 6.2: Simulation parameters

Parameter	Value
Simulation duration	$3600s$
Simulated area	300m x 300m, nodes placed randomly in a circle sink in the center
Number of nodes	50
Number of bottlenecks advertised	8
Load balance step δ	0.1 (10%)
Traffic type, rate	CBR, 1 pkt/min
Data packet size	127 bytes (incl. MAC headers)
RPL	MinHopRankIncrease = 128
Trickle	$Imin = 2^7 ms$, $Imax = 16$, $k = 10$
MAC layer	802.15.4 beacon mode
MAC parameters	BO=7, S=2
Energy consumption	CC2420 datasheet

- our multiparent version of RPL: a node forwards the traffic fairly to all its parents, based on the lifetime of their bottlenecks.

We also compared the following routing metrics:

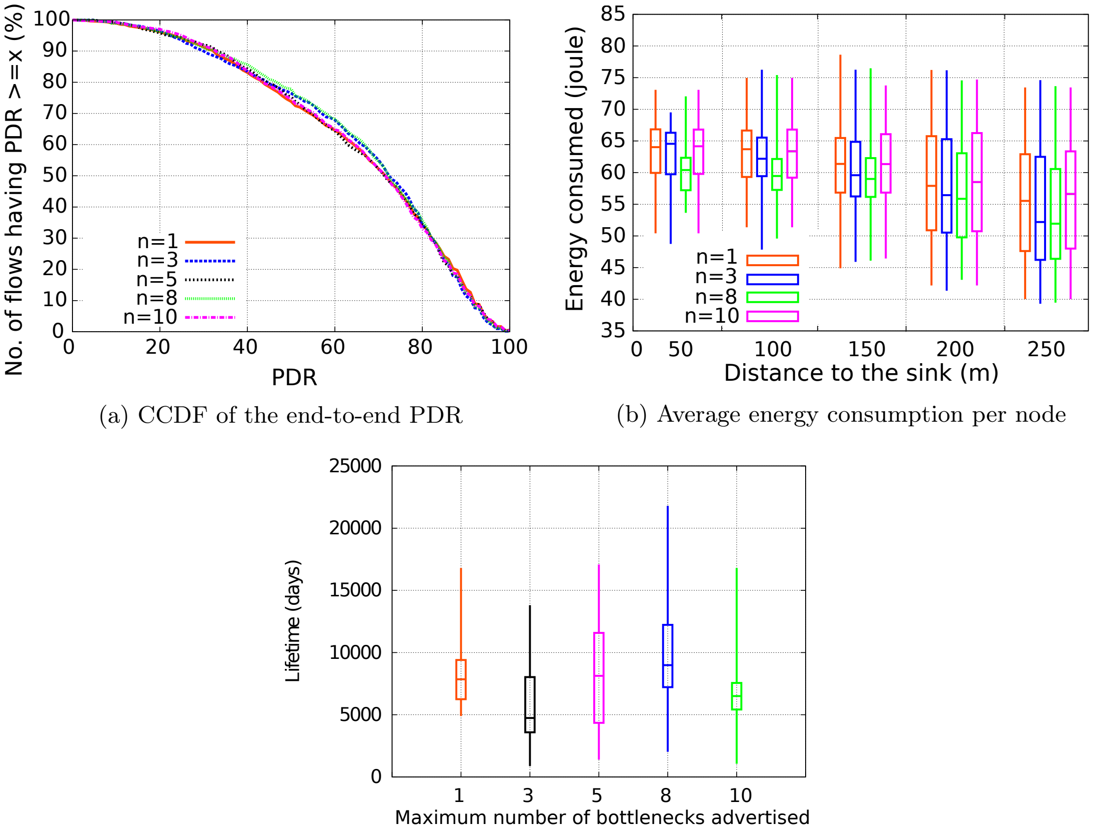
- the residual energy: the remaining energy of a node, as implemented by Kamgueu *et al.* [Kam+12];
- ETX (using the Minimum Rank with Hysteresis Objective Function [Gna12]): the average number of transmissions for an acknowledged packet;
- energy: a linear combination of ETX and the residual energy, as proposed by Chang *et al.* [Cha+13];
- ELT: our metric, which directly exploits the remaining lifetime of a node.

We are interested in studying the following aspects:

1. the impact of different parameters of the multiparent version (e.g., number of bottlenecks advertised) on the network performance;
2. a performance evaluation and comparison within the different versions of RPL and routing metrics:
   - network reliability, measured as the end-to-end packet delivery ratio (PDR), i.e., ratio of packets received by the sink;
   - end-to-end delay: the delay between the packet generation and its reception by the sink (considered only for delivered packets);
   - energy consumption: the energy consumed by the nodes during the whole simulation period;
   - network lifetime: time before the first node dies when considering only the energy consumed by the CC2420 chipset.
3. the dynamics of the network, measured as the number of preferred parent changes.

6.6.1 Tuning the Parameters

First, we study the impact of different parameters of the multiparent version on the performance of the network.



(a) CCDF of the end-to-end PDR

(b) Average energy consumption per node

(c) Expected Lifetime
(time until the first node dies)

Figure 6.6: Impact of the no. of bottlenecks (n) included in the DIO

Number of Bottlenecks Advertised

First, we have investigated the impact of the number of bottlenecks included in the DIO. A too small number means a node under-estimates the impact of its traffic on the other bottlenecks.

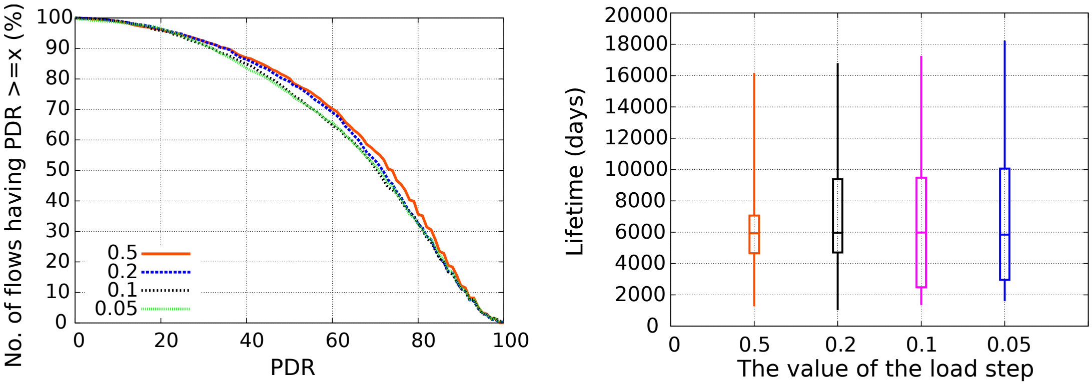
In Figure 6.6a we plotted the CCDF of the end-to-end PDR for all the flows, in function of the maximum number of bottlenecks advertised by a node. We can see that the PDR is almost the same, no matter how many bottlenecks a node advertises. Indeed, a node considers both its ELT and the ELT of the bottlenecks. Since ELT accounts for the ETX, it implicitly takes reliability into account. The number of bottlenecks has rather an impact on the energy consumption.

In Figure 6.6b we plotted the energy consumed by the nodes during the whole simulation, against their physical distance from the sink (i.e., the border router). No matter the number of bottlenecks advertised, our solution manages to construct an energy balanced topology. Our solution is hopefully not too sensitive to the exact number of bottlenecks to advertise. A small number of bottlenecks is sufficient to even balance the energy consumption in the network.

Here, we see that adverting 8 bottlenecks represents the optimal value. Without increasing too much the DIO size, we manage to not underestimate the consumption of the *secondary* bottlenecks.

Finally, Figure 6.6c illustrates the impact of the number of bottlenecks on the network lifetime. With only one bottleneck we manage to estimate quite accurately the network lifetime. Besides, only one bottleneck means we reduce the number of variables to maintain to estimate the ELT of each path: we reduce consequently the instability.

However, increasing the number of bottlenecks tends to also balance more finely the energy



(a) CCDF of the end-to-end PDR

(b) Expected Lifetime (time until the first node dies)

Figure 6.7: Impact of the parameter δ

consumption along all the paths. This finer estimation tends to counter-balance the number of parent changes. Again, 8 represents here the optimal value for the number of bottlenecks to include in the `DIO`.

Impact of δ on the Network Performance

We finally studied the impact of the load balancing parameter δ which is used to compute the traffic to send to each parent.

We can see in Figure 6.7a that it has no impact on the end-to-end packet delivery ratio. Indeed, the multipath ELT succeeds to find reliable routes, even if the energy is not so well balanced (too small δ value).

Moreover, when δ is sufficiently small, it has no impact on the lifetime of the network. We can notice in Figure 6.7b that when $\delta = 0.5$ the lifetime is slightly smaller. Since in most of the cases a node has more than two parents, it will be more difficult for the algorithm to optimally distribute the last 0.5% of the traffic. This sub-optimal distribution negatively impacts the lifetime.

6.6.2 Comparison of the Different Routing Metrics and RPL Versions

We offer in this section a thorough performance evaluation of our proposed multiparent version of RPL, as well as a comparison with the other routing metrics: the residual energy, ETX and the *energy*.

Packet Delivery Ratio

Figure 6.8 illustrates the complementary cumulative distribution function (CCDF) of the PDR for all the flows. We can observe that the residual energy presents the worst reliability: some nodes with bad radio links are selected only because they have a large residual energy. These bad links have a negative impact on the packet delivery ratio.

At the opposite end, ETX presents the highest reliability: only the best links are used to route the packets. The metric *energy* represents as expected, a tradeoff between the residual energy and the ETX metrics.

Finally, our multiparent version of ELT achieves almost the same reliability as ETX. Our metric selects routes with the largest residual energy, without impacting negatively the reliability. Some of the nodes, however, will select a slightly less good radio link, but do so by preserving the network lifetime.

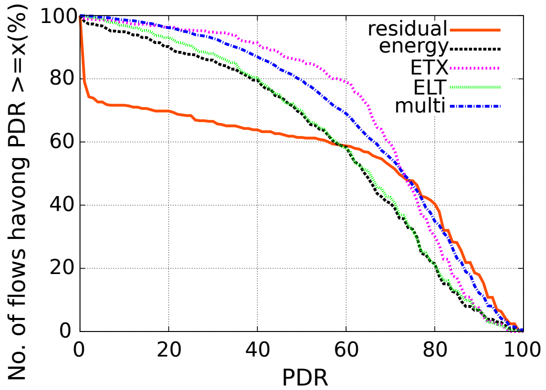


Figure 6.8: CCDF of the end-to-end PDR

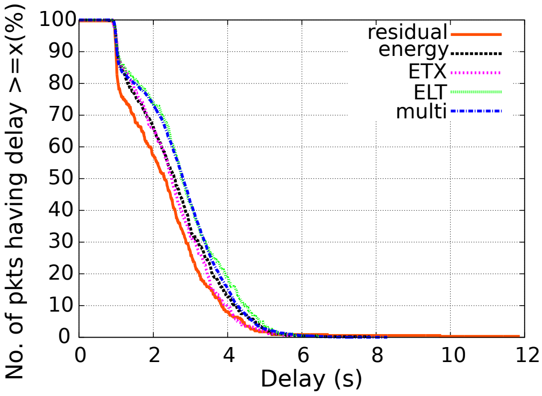


Figure 6.9: CCDF of the end-to-end delay

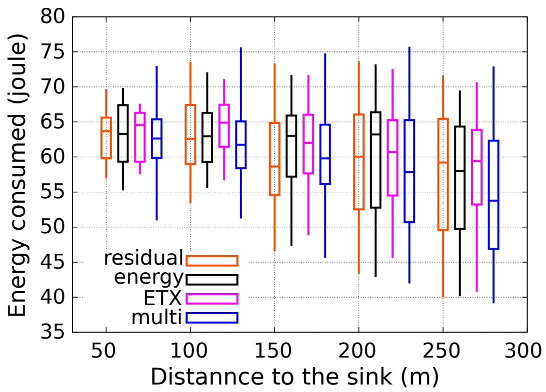


Figure 6.10: Energy consumption of nodes in function of their physical distance to the sink

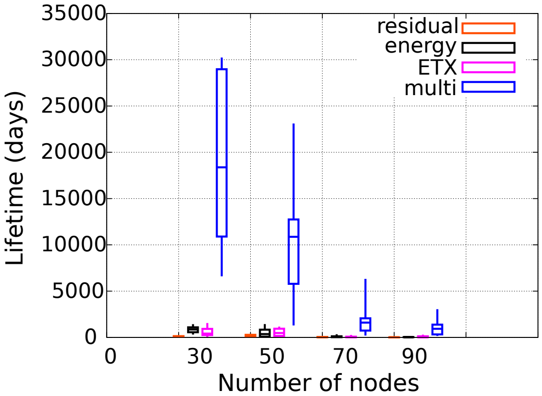


Figure 6.11: Network Lifetime (time until the first node dies) in function of the density

End-to-end Delay

Figure 6.9 illustrates the CCDF of the end-to-end delay for of all the received packets. This delay is quite insensitive to the metric used for routing.

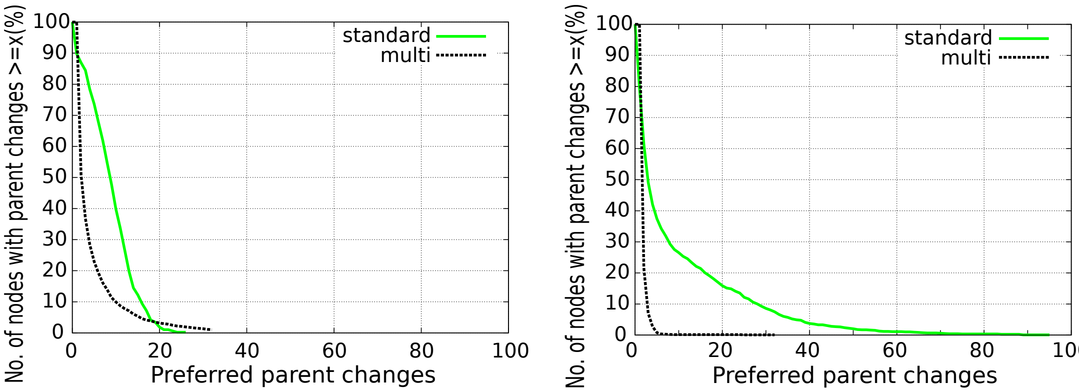
Indeed, the limited number of retransmissions at the MAC layer makes that bad radio links tend to reduce the reliability (i.e., the packet is dropped) while having a marginal impact on the delay. In particular, the retransmission delay is practically much shorter than the buffering delay: since the network operates at low duty-cycle ratios, a packet is buffered for a long time before the node wakes up and sends it to one of its parents.

Energy Efficiency

Figure 6.10 represents the box plot of the average energy consumed by the nodes during the whole simulation, against their physical distance from the sink (i.e., the border router).

The multiparent version with the ELT has the lowest energy consumption among all the routing metrics. The residual energy seems to be quite energy efficient, as well. However, it also presents the worse reliability. Since less packets are forwarded, they also consume less energy.

ETX on the other hand, is the most energy inefficient metric. Still, this cannot be explained only by the fact that it offers the best reliability. We have to remember that the multiparent ELT has similar PDR, while consuming less energy. Most surely this increase in the energy consumption is due to a larger overhead.



(a) Using ELT

(b) Using ETX

Figure 6.12: CCDF of the number of preferred parent changes during the whole simulation

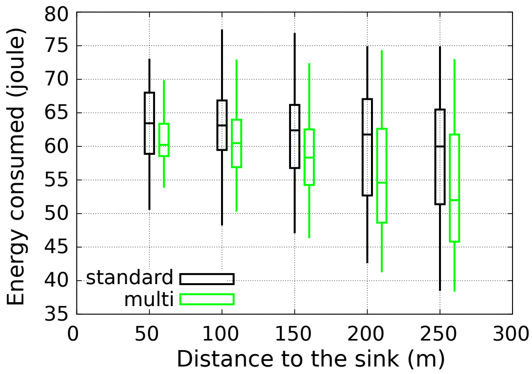


Figure 6.13: Energy consumption in function of the physical distance of the nodes from the sink

Network Lifetime

Finally, we evaluated the network lifetime in function of the density. We increased the number of nodes within the same simulation area to isolate the impact of the density.

We can observe in Figure 6.11 that multipath ELT clearly outperforms the standard RPL. Multipath routing helps balancing more accurately the energy: routing decisions are not binary, and the traffic is spread to all the bottlenecks. Besides, our strategy of re-allocating the traffic to each parent based on their lifetime, is conservative and avoids sudden redirections.

We can notice that even though multipath ELT has the largest energy consumption in the worst case scenario (Figure 6.10), it still manages to have the best lifetime (Figure 6.11). This is due to the fact that the algorithm consumes more energy during the bootstrap period, but preserves lifetime better on the long run.

6.6.3 Network Dynamics

We have measured the stability of the network as the number of preferred parent changes. We count the number of parent changes during 1 hour of simulation, and plot the associated CCDF.

We can see in Figure 6.12a that when ELT is used, the standard version exhibits high instability: most of the nodes frequently change their parents. During the simulation, one half of the nodes change at least 8 times their preferred parent. For each change, the trickle timer is reset and the `DIO` are transmitted more frequently.

On the contrary, the multipath version significantly improves the stability. By forbidding sudden parent weight changes, a node smoothens the traffic redirection. More than 80% of the nodes change at most 4 times their preferred parent.

We also wanted to evaluate the network dynamics for when the ETX is used as the routing metric. In consequence, we implemented the multiparent version of RPL for ETX too. We can see the results for the number of preferred parent changes in Figure 6.12b. Indeed, using multiple parents for routing significantly improves the stability of the routing topology.

The decrease in the number of parent changes also means a decrease of the number of `DIO` sent in the network, and hence of the energy consumption. We can see this effect in Figure 6.13 that presents the box plot of the energy consumed during the whole simulation, in function of the physical distance of the nodes from the sink.

6.7 Conclusion

We proposed an energy-balanced version of RPL. First, we constructed the DODAG based on the ELT metric, which accurately estimates the lifetime of all the routes toward the border router. By selecting as parents the nodes with the strongest paths (i.e., larger ELT), we improve the network lifetime.

Second, we proposed a multipath approach to fully exploit the DAG structure. A node exploits all its parents, assigning a weight of traffic to each of them. In this way, a node fairly distributes the energy consumption among all the paths toward the border router. Since each node receives fairly a quantity of traffic to forward, energy consumption is well balanced.

Finally, a preferred parent is removed only when it becomes useless (i.e., it forwards no traffic). We also dealt efficiently with inaccuracies in the metric estimation. Indeed, the radio link quality is stochastic, and the routes constructed by RPL should not change if the radio link quality has not *significantly* changed. We manage to limit the number of parent changes and thus, to reduce the energy consumption of the nodes.

As further work, it will be interesting to investigate the conditions that make the energy distribution more efficient with a small number of bottlenecks. Then, we could study how a node may dynamically select the bottlenecks to include in its `DIO`, instead of limiting them by a fixed value.

Also, we only took into consideration time-driven applications, with CBR traffic. The burst traffic used in event-driven applications is unpredictable and can have a negative impact on the convergence of the protocol. Since ELT estimates the lifetime of a node in the current traffic conditions, it does not account for this extra traffic. A tradeoff will have to be made between maintaing the stability in the network and maximizing the energy consumption.

Chapter 7

Conclusion and Perspectives

This chapter concludes the thesis, reminding the addressed problem, highlighting the contributions, and opening up perspectives.

7.1 Conclusion

The goal of this thesis was to solve some key problems of the protocols recently standardized for the Internet of Things. Since the communication between devices is the building block of any network, we focused on the IEEE 802.15.4 MAC protocol and on RPL, the IPv6 Routing Protocol for Low-power and Lossy Networks.

We started by enhancing the beacon mode of IEEE 802.15.4 with the Contention Broadcast Only Period approach. At the beginning of each superframe, a coordinator waits a randomly chosen period of time before sending its beacon, followed by the enqueued broadcast packets. This mechanism allows several coordinators to send their beacons and broadcast packets in the same superframe. Since our approach does not depend on a fixed slot duration in which to send the beacon (like the Beacon Only Period mechanism), we manage to reduce bandwidth wastage.

To guarantee a reliable delivery of the broadcast packets in lossy environments, we proposed the use of broadcast sequence numbers. The simulations with a realistic physical model showed that our approach manages to successfully deliver broadcast packets, while inducing a low overhead in the network.

Then, in Chapter 4, we evaluated the impact of different link quality metrics on RPL. We first isolated an instability problem: whatever the routing metric is, a node changes its preferred parent. These oscillations generate a large overhead because of the trickle algorithm: the trickle timer is reset, triggering a more frequent `DIO` transmission. These control packets waste energy.

We also highlighted the existence of a tradeoff between stability and efficiency for the existing metrics. For instance, the hop count metric exhibits the lowest instability, but performs very poorly because it tends to use bad radio links. LQI also limits the instability, but offers larger end-to-end delays. Oppositely, ETX balances more efficiently the load among the different nodes but maximizes the number of trickle timer resets. Clearly, a new routing metric needed to be proposed that exhibits a stable behavior, while reflecting well the radio link quality.

In consequence, we proposed in Chapter 5 a new routing metric that efficiently balances the load among the different nodes, while keeping a low overhead, and avoiding instabilities: the Expected Lifetime (ETX). This metric estimates the time at which a node will run out of energy in the current traffic conditions, according to its residual energy and the link quality to its preferred parent. Combined with RPL, ETX manages to maximize the network lifetime by identifying the bottlenecks of the network and creating energy balanced paths.

We also showed through simulations that ETX has performance results close to ETX in terms of reliability and delay, while consuming less energy. Moreover, ETX clearly maximizes the network lifetime, no matter the density of the network.

In Chapter 6, we extended this metric to the multipath scenario in RPL. We constructed a Directed Acyclic Graph (DAG) based on the ETX metric, which accurately estimates the lifetime of all the routes toward the border router. By selecting as parents the nodes with the strongest paths (i.e., largest ETX), we improved the network lifetime.

To fully exploit the DAG structure, we proposed that a node forwards its traffic to all of its parents, such that each path consumes the same quantity of energy. We also dealt with the inaccuracies in the metric estimation by progressively redirecting the traffic when recomputing the weights for each parent.

The simulation results showed that we considerably improved the lifetime of the network. Moreover, our multiparent approach reduces the instability in the routing topology, both when ETX and ETX are used as routing metrics.

7.2 Perspectives

The contributions of this thesis can be extended in several directions. Let us now present some of them.

7.2.1 Experiments

One of the directions for continuing the works presented in this thesis is a validation through experiments. While we systematically used a realistic physical model with shadowing in our simulations, experiments would allow to reveal more interesting details of this research.

In Chapter 4 we highlighted an instability problem of the routing topology constructed with RPL. Indeed, RPL seems to not converge to a stable list of routes, even when the nodes are static, the traffic is CBR, and the radio conditions remain unchanged. A fortiori, the presence of asymmetrical links, and a more fast varying radio channel from the experimental/real conditions will amplify the problem rather than solving it.

A faster varying radio channel could also have an impact on the convergence of the ETX routing metric, since it uses the current traffic conditions to compute the expected lifetime of a node. Moreover, in the case of the multipath version, the algorithm used for traffic redirection to smoothen the variations, could delay the convergence.

Regarding the presence of asymmetrical links, ETX exploits ETX for the estimation of the link quality and thus, should be able to handle them. We recall that ETX accounts for the PDR in both directions of a link. Hence, it suffices to have an accurate implementation of ETX.

7.2.2 Link Quality Estimation

ETX is one of the most used metrics for the computation of the link quality. However, as it is always searching for the best instantaneous link quality, a node often changes its next hop, inducing instability in the network. Indeed, just a small variation in the link quality estimation is sufficient to trigger the change of the next hop.

These instabilities may be related to the computation of the metric, which represents just an instantaneous value, or to the evaluation of the metric over a period of time, through an estimation technique.

A good estimator should allow the metric to be reactive to persistent changes, yet stable in front of transient (short-term) variations. Several estimation techniques have been proposed in the literature to smoothen the values of the link quality metrics. Among them, the Exponential Weighted Moving Average (EWMA) proved to outperform all of them [Bac+12].

EWMA computes the average of the instantaneous values over a period of time, by giving different weights to the past and present values. However, the choice of these weights has a direct impact on the stability and reactivity of the link quality metric.

In our simulations, we implemented ETX using the EWMA estimator, favoring the stability of the metric when choosing the value of the weights. Still, we observed a lot of instability in the network, especially when we used RPL with IEEE 802.15.4-2006 non beacon, which uses a peer-to-peer topology (i.e., a more meshed topology than the cluster-tree).

Since rerouting is very energy and time consuming, a new estimation technique should be proposed that can tolerate short-term variations in the metric, while being reactive to persistent changes.

7.2.3 Energy Balancing with Multiple Instances

The Internet of Things is composed of several networks of computers, devices, and objects with communication capabilities. In such a heterogenous environment, where networks overlap, different applications could share common resources. For example, in a city, the same network can be used to collect information from water meters, environmental data (CO2 levels, temperature, etc.), and traffic.

In order to support different applications, a network can run multiple RPL instances. A border router constructs a DODAG in an instance using an objective function. Each instance is optimized for the specific constraints of the application, e.g., minimum latency, high reliability, etc. More specifically, a network can support several DODAGs, each of them constructed in a different instance, using its own objective function and routing metric.

With such a scenario in mind, let us suppose now that only one DODAG is constructed using the Expected Lifetime metric, and focus on this instance. The other DODAGs can use any of the routing metrics and objective functions defined in companion RFCs. ETX uses information about the quality of the links and the quantity of traffic a node has to forward to estimate its lifetime. The preferred parent of a node in this instance might not be its preferred parent in the other instances. Since ETX considers that all the traffic is sent to the preferred parent, it might underestimate or overestimate the time at which a node runs out of energy. While we could implement more statistics at the level of each node, this might not always be possible. Also, the situation becomes more complicated in the multipath scenario, where a node has to compute the proportion of traffic that it has to sent to each of its parents.

7.2.4 Loop Detection and Avoidance

RPL constructs a loop-free topology by forbidding a node to process a `DIO` from a neighbor advertising a rank higher than itself. RPL also disallows greediness, i.e., a node cannot increase its rank in order to improve some metric or to increase the size of its parent set. Still, loops may appear in time because of the variations in the physical connectivity.

In practice, RPL does not guarantee loop-free path selection, but it can *detect and repair a loop as soon as it is used* [Win+12]. Indeed, when a node detects a loop, it institutes a local repair operation. However, RPL does not specify such a mechanism.

One example of a method that can be used as a local repair is the route poisoning mechanism. A node detaches from the DODAG, advertises a rank with `INFINITE_RANK` (i.e., it poisons its routes so that the nodes in its sub-DODAG deletes it from their parent set), and then reattaches to the DODAG. This is quite an aggressive mechanism and can leave the network disconnected. Also, it induces a lot of overhead and instability in the network. All the nodes that have as preferred parent the node that poisoned its routes will have to re-choose their preferred parent, or initiate a local repair, if no backup parents are present. Clearly, more efficient mechanisms have to be proposed.

Appendix A

A.1 Correctness: $(1+\delta)$ - approximation

Theorem 6.1. *The presented greedy algorithm is a $1+\delta$ approximation.*

Proof. Let t_{NB} be the traffic from a node N to a bottleneck B, k the maximum number of parents that a node can have, $\overline{x}$ the notation for the real value of x, and x_{opt} the notation for the optimal value for x, $\forall x$. Table 6.1 from page 71 depicts the rest of the notation.

The fact that the greedy algorithm is a $1+\delta$ approximation means that the traffic that reaches a bottleneck is at most the optimal_value $\times$ $(1+\delta)$. Hence, the theorem is equivalent to:

$$\forall B_i \in Bottlenecks : \sum_{\forall N_j \in Network} \overline{t_{N_j B_i}} \leq (1+\delta) \sum_{\forall N_j \in Network} t_{N_j B_i \ opt} \tag{A.1}$$

Let us take first the specific case when just one node in the network (N) has packets to send. The reasoning holds also for when there is more than just one node generating packets in the network. If each node is at $(1+\delta)$ from the optimal, globally it will also be at $(1+\delta)$ from the optimal. For reasons of too many variables (which leads to complicated notations), we decided to not present this part of the proof. Moreover, it is very similar to the case of a single node generating packets.

If N is the only node in the network generating packets, the Equation A.1 is equivalent to:

$$\forall B_i \in Bottlenecks : \overline{t_{NB_i}} \leq t_{NB_i \ opt} + \delta \times T_N \tag{A.2}$$

Let us assume that the last δ of the traffic is sent by N to the parent P_1, which is not optimal. If we focus on the most constrained bottleneck (e.g., B_1), we distinguish the following cases:

1. There are other parents of N, besides P_1, that forward traffic to B_1:
   $\exists P_i \in \{Parents(N) \setminus P_1\}$ s.t. $r_{Pi,B1} > 0 \Rightarrow$

$$\overline{t_{N_1 B_1}} = \sum_{P_i \in Parents(N)} \overline{\alpha_i} \times r_{P_i,B_1} \times T_N \tag{A.3}$$

   Let $Parents^*(N) = \{Parents(N) \setminus P_1 \mid r_{Pi,B1} > 0\}$. Then:

$$\overline{t_{N_1 B_1}} = \overline{\alpha_1} \times r_{P_1,B_1} \times T_N + \sum_{P_i \in Parents^*(N)} \overline{\alpha_i} \times r_{P_i,B_1} \times T_N \tag{A.4}$$

   Adding δ traffic w.r.t. the optimal to parent P_1 means:

$$\overline{\alpha_1} \leq \alpha_{1 \ opt} + \delta \tag{A.5}$$

Since we remove from the rest of the parents the ratio of traffic that is not sent compared to the optimal case (which is smaller than δ), we obtain the following relation:

$$\sum_{P_i \in Parents^*(N)} \overline{\alpha_i} \leq \sum_{P_i \in Parents^*(N)} \alpha_{i\ opt} - \delta \tag{A.6}$$

Now, if we replace the values of $\overline{\alpha_i}$ from the previous inequalities (A.5 and A.6) in Equation A.4, we obtain the following inequality:

$$\overline{t_{NB_1}} \leq (\alpha_{1\ opt} + \delta) \times r_{P_1,B_1} \times T_N + \left(\sum_{P_i \in Parents^*(N)} \alpha_{i\ opt} - \delta \right) \times r_{P_i,B_1} \times T_N \tag{A.7}$$

$$\overline{t_{NB_1}} \leq \sum_{P_i \in Parents(N)} \alpha_{i\ opt} \times r_{P_i,B_1} \times T_N + \left(r_{P_1 B_1} - \sum_{P_i \in Parents(N)} r_{P_i,B_1} \right) \times \delta \times T_N \tag{A.8}$$

$$\overline{t_{NB_1}} \leq t_{NB_1 opt} + \delta \times T_N \tag{A.9}$$

Since N is the only node in the network generating packets, we obtain c.f. Equation A.1:

$$\sum_{\forall N_j \in Network} \overline{t_{N_j B_i}} \leq (1+\delta) \sum_{\forall N_j \in Network} t_{N_j B_i\ opt} \tag{A.10}$$

2. P_1 is the only parent of N forwarding traffic to the bottleneck B_1:
   $\forall P_i \in \{Parents(N) \setminus P_1\} : r_{P_i,B_1} = 0 \Rightarrow$

   $$\overline{t_{N_1 B_1}} = \overline{\alpha_1} \times r_{P_1,B_1} \times T_N = \overline{\alpha_1} \times T_N \tag{A.11}$$

   Now, if we replace $\overline{\alpha_1}$ by $\alpha_{1\ opt} + \delta$ in Equation A.11 we obtain:

   $$\overline{t_{N_1 B_1}} \leq (\alpha_{1\ opt} + \delta) \times r_{P_1,B_1} \times T_N \leq t_{N_1 B_1 opt} + \delta \times T_N \tag{A.12}$$

   Again, since N is the only node in the network generating packets, we obtain c.f. Equation A.1:

$$\sum_{\forall N_j \in Network} \overline{t_{N_j B_i}} \leq (1+\delta) \sum_{\forall N_j \in Network} t_{N_j B_i\ opt} \tag{A.13}$$

□

Bibliography

[802] *IEEE 802.15 WPAN Task Group 4b (TG4b)*. `http://www.ieee802.org/15/pub/TG4b.html.2006`.

[Acc+11] N. Accettura, L.A. Grieco, G. Boggia, and P. Camarda. “Performance Analysis of the RPL Routing Protocol”. In: *ICM*. IEEE. 2011.

[Ady+04] A Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou. “A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks”. In: *BROADNETS*. IEEE, 2004.

[Aky+02a] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. “A Survey on Sensor Networks”. In: *Communications Magazine, IEEE* 40.8 (2002), pp. 102–114.

[Aky+02b] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. “Wireless Sensor Networks: A Survey”. In: *Computer Networks, Elsevier* 38.4 (2002), pp. 393–422.

[AP14] N. Accettura and G. Piro. “Optimal and Secure Protocols in the IETF 6TiSCH Communication Stack”. In: *ISIE*. IEEE, 2014.

[AY05] K. Akkaya and M. Younis. “A Survey on Routing Protocols for Wireless Sensor Networks”. In: *Ad Hoc Networks, Elsevier* 3.3 (2005), pp. 325 –349.

[Bac+10a] A. Bachir, F. Theoleyre, A. Duda, and K. K. Leung. “Energy-Efficient Broadcasts in Wireless Sensor Networks with Multiple Virtual Channels”. In: *WCNC*. IEEE. 2010.

[Bac+10b] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung. “MAC Essentials for Wireless Sensor Networks”. In: *Communications Surveys and Tutorials, IEEE* 12.2 (2010), pp. 222–248.

[Bac+12] N. Baccour, A. Koubaa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves. “Radio Link Quality Estimation in Wireless Sensor Networks: A Survey”. In: *Transactions on Sensor Networks, ACM* 8.4 (2012), 34:1–34:33.

[Bar+08] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange. “SensorScope: Out-of-the-Box Environmental Monitoring”. In: *IPSN*. IEEE, 2008.

[BBP10] A. Brandt, J. Buron, and G. Porcu. *Home Automation Routing Requirements in Low-Power and Lossy Networks*. RFC 5826. IETF, 2010.

[Bue+06] M. Buettner, G. V. Yee, E. Anderson, and R. Han. “X-MAC: A Short Preamble MAC Protocol for Duty-cycled Wireless Sensor Networks”. In: *SenSys*. ACM, 2006.

[CDG06] A. Conta, S. Deering, and M. Gupta. *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*. RFC 4443. IETF, 2006.

[Cer+05] A. Cerpa, J. L. Wong, M. Potkonjak, and D. Estrin. “Temporal Properties of Low Power Wireless Links: Modeling and Implications on Multi-hop Routing”. In: *MobiHoc*. ACM, 2005.

[Cer+09] M. Ceriotti, L. Mottola, G. P. Picco, A. L. Murphy, S. Guna, M. Corra, M. Pozzi, D. Zonta, and P. Zanon. “Monitoring Heritage Buildings with Wireless Sensor Networks: The Torre Aquila Deployment”. In: *IPSN*. IEEE, 2009.

[Cha+13] L.-H. Chang, T.-H. Lee, S.-J. Chen, and C.-Y. Liao. “Energy-Efficient Oriented Routing Algorithm in Wireless Sensor Networks”. In: *SMC*. IEEE. 2013.

[Chi+06] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher. “Real-time Power-Aware Routing in Sensor Networks”. In: *IWQoS*. IEEE. 2006.

[Chi+10] O. Chipara, C. Lu, T. C. Bailey, and G.-C. Roman. “Reliable Clinical Monitoring using Wireless Sensor Networks: Experiences in a Step-down Hospital Unit”. In: *SenSys*. ACM, 2010.

[CHP11] T. Clausen, U. Herberg, and M. Philipp. “A critical evaluation of the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL)”. In: *WiMob*. IEEE. 2011.

[Cis] *The Internet of Things Infographic*. URL: `http://blogs.cisco.com/news/the-internet-of-things-infographic/`.

[CK03] C. Y. Chong and S. P. Kumar. “Sensor Networks: Evolution, Opportunities, and Challenges”. In: *Proceedings of the IEEE* 91.8 (2003), pp. 1247–1256.

[Cla+14] T. Clausen, J. Yi, A. Niktash, I. Igarashi, H. Satoh, U. Herberg, C. Lavenu, T. Lys, and J. Dean. *The Lightweight On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng) draft-clausen-lln-loadng-11*. Internet-Draft. IETF, 2014.

[Cm5] *CM500 Wireless Mote, produced by Advanticsys*. URL: `http://www.advanticsys.com/`.

[CN06] Y. Chen and N. Nasser. “Energy-Balancing Multipath Routing Protocol for Wireless Sensor Networks”. In: *QShine*. ACM, 2006.

[Col+13] U. M. Colesanti, A. Lo Russo, M. Paoli, C. Petrioli, and A. Vitaletti. “Structural Health Monitoring in an Underground Construction Site: The Roman Experience”. In: *SenSys*. ACM, 2013.

[Con] *Contiki: The Open Source OS for the Internet of Things*. http://www.contiki-os.org/.

[Cou+05] D. De Couto, D. Aguayo, J. Bicket, and R. Morris. “A High-Throughput Path Metric for Multi-hop Wireless Routing”. In: *Wireless Networks, ACM* 11.4 (2005), pp. 419–434.

[CS03] V.S. Cronin and K.A. Sverdrup. “Defining Static Correction for Jointly Relocated Earthquakes Along the Blanco Transform Fault Zone Based on SOSUS Hydrophone Data”. In: *Proceedings of Oceans*. IEEE, 2003.

[CT04] J.-H. Chang and L. Tassiulas. “Maximum Lifetime Routing in Wireless Sensor Networks”. In: *Transactions on Networking, IEEE/ACM* 12.4 (2004), pp. 609–619.

[CT11] Y. Chen and A. Terzis. “On the Implications of the Log-normal Path Loss Model: An Efficient Method to Deploy and Move Sensor Motes”. In: *SenSys*. ACM. 2011.

[DDB12] S. Dawans, S. Duquennoy, and O. Bonaventure. “On Link Estimation in Dense RPL Deployments”. In: *LCN*. IEEE. 2012.

[Deh+08] A. Dehghani, K. Jamshidi, S. Mirshams, and K. RahimiZadeh. “A Routing Metric for Load Balancing Considering Energy and Delay Constraints in Wireless Sensor Network”. In: *IST*. IEEE, 2008.

[Dev+11] P. Devis, T. Clausen, J. Hui, O. Gnawali, and J. Ko. *The Trickle Algorithm*. RFC 6206. IETF, 2011.

[DF+06] H. Dubois-Ferriere, R. Meier, L. Fabre, and P. Metrailler. "TinyNode: A Comprehensive Platform for Wireless Sensor Network Applications". In: *IPSN*. ACM/IEEE. 2006, pp. 358–365.

[Din+06] G. Ding, Z. Sahinoglu, P. Orlik, J. Zhang, and B. Bhargava. "Tree-Based Data Broadcast in IEEE 802.15.4 and ZigBee Networks". In: *Transactions on Mobile Computing, IEEE* 5.11 (2006), pp. 1561 –1574.

[DL03] T. van Dam and K. Langendoen. "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks". In: *SenSys*. ACM, 2003.

[DLV13] S. Duquennoy, O. Landsiedel, and T. Voigt. "Let the Tree Bloom: Scalable Opportunistic Routing with ORPL". In: *SenSys*. ACM. 2013.

[Doh+09] M. Dohler, T. Watteyne, T. Winter, and D. Barthel. *Routing Requirements for Urban Low-Power and Lossy Networks*. RFC 5548. IETF, 2009.

[DPZ04] R. Draves, J. Padhye, and B. Zill. "Routing in Multi-radio, Multi-hop Wireless Mesh Networks". In: *MobiCom*. ACM, 2004.

[Dsn] *Distributed Sensor Networks*. Semiannual Technical Summary Report. Massachusetts Institute of Technology, Lincoln Laboratory, 1987.

[Dun11] A. Dunkels. *The ContikiMAC Radio Duty Cycling Protocol*. Technical Report T2011:13. Swedish Institute of Computer Science, 2011.

[Dyo+10] V. Dyo et al. "Evolution and Sustainability of a Wildlife Monitoring Sensor Network". In: *SenSys*. ACM, 2010.

[EHD04] A. El-Hoiydi and J.-D. Decotignie. "WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks". In: *ISCC*. IEEE, 2004.

[EV06] S. C. Ergen and P. Varaiya. "PEDAMACS: Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks". In: *Transactions on Mobile Computing, IEEE* 5.7 (2006), pp. 920–930.

[Eva11] D. Evans. *The Internet of Things How the Next Evolution of the Internet Is Changing Everything*. White Paper. CISCO, 2011.

[FLE06] E. Felemban, C.-G. Lee, and E. Ekici. "MMSPEED: Multipath Multi-SPEED Protocol for QoS Guarantee of Reliability and Timeliness in Wireless Sensor Networks". In: *Transactions on Mobile Computing, IEEE* 5.6 (2006), pp. 738–754.

[Fon+07] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis. "Four-Bit Wireless Link Estimation". In: *HotNets*. ACM, 2007.

[FPH05] J. Faruque, K. Psounis, and A. Helmy. "Analysis of Gradient-Based Routing Protocols in Sensor Networks". In: *DCOSS*. IEEE. 2005.

[Gad+12] O. Gaddour, A Koubaa, S. Chaudhry, M. Tezeghdanti, R. Chaari, and M. Abid. "Simulation and Performance Evaluation of DAG Construction with RPL". In: *ComNet*. IEEE. 2012.

[Gan+01] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. "Highly-resilient, Energy-efficient Multipath Routing in Wireless Sensor Networks". In: *SIGMOBILE, ACM* 5.4 (2001), pp. 11–25.

[Gha+14] E. Ghadimi, O. Landsiedel, P. Soldati, S. Duquennoy, and M. Johansson. "Opportunistic Routing in Low Duty-Cycle Wireless Sensor Networks". In: *Transactions on Sensor Networks, ACM* 10.4 (2014), 67:1–67:39.

[GK12] O. Gaddour and A. Kouba. "RPL in a Nutshell: A Survey". In: *Computer Networks, Elsevier* 56.14 (2012), pp. 3163–3178.

[Gna+09] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. "Collection Tree Protocol". In: *SenSys*. ACM, 2009.

[Gna12] Gnawali, O. and Levis, P. *The Minimum Rank with Hysteresis Objective Function*. RFC 6719. IETF, 2012.

[Goy+13] M. Goyal, E. Baccelli, M. Philipp, A. Brandt, and J. Martocci. *Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks*. RFC. IETF, 2013.

[Gro99] N. Gross. "21 Ideas for the 21st Century". In: *BusinessWeek* (1999).

[Gua+14] L. Guan, K. Kuladinithi, T. Potsch, and C. Goerg. "A Deeper Understanding of Interoperability between TinyRPL and ContikiRPL". In: *ISSNIP*. IEEE. 2014.

[Guo+09] S. Guo, Y. Gu, B. Jiang, and T. He. "Opportunistic Flooding in Low-duty-cycle Wireless Sensor Networks with Unreliable Links". In: *MobiCom*. ACM. 2009.

[HC11a] U. Herberg and T. Clausen. "A Comparative Performance Study of the Routing Protocols LOAD and RPL with Bi-directional Traffic in Low-power and Lossy Networks (LLN)". In: *PE-WASUN*. Miami, Florida, USA: ACM, 2011.

[HC11b] K.-S. Hong and L. Choi. "DAG-based Multipath Routing for Mobile Sensor Networks". In: *ICTC*. IEEE, 2011.

[HCG08] E.B. Hamida, G. Chelius, and J.-M. Gorce. "On the Complexity of an Accurate and Precise Performance Evaluation of Wireless Networks using Simulations". In: *MSWiM*. ACM. 2008.

[HEB08] M. Hamdi, N. Essaddi, and N. Boudriga. "Energy-Efficient Routing in Wireless Sensor Networks Using Probabilistic Strategies". In: *WCNC*. IEEE, 2008.

[HG13] D. Han and O. Gnawali. "Performance of RPL under Wireless Interference". In: *Communications Magazine, IEEE* 51.2 (2013), pp. 137–143.

[HMA13] K. Heurtefeux, H. Menouar, and N. AbuAli. "Experimental Evaluation of a Routing Protocol for WSNs: RPL Robustness Under Study". In: *WiMob*. IEEE. 2013.

[HSF14] AK.M.M. Hossain, C.J. Sreenan, and S. Fedor. "A Neighbour Disjoint Multipath Scheme for Fault Tolerant Wireless Sensor Networks". In: *DCOSS*. IEEE, 2014.

[HT11] J. Hui and P. Thubert. *Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks*. RFC 6282. IETF, 2011.

[HTK04] X. Hou, D. Tipper, and J. Kabara. "Label-based Multipath Routing (LMR) in Wireless Sensor Networks". In: *ISART*. 2004.

[Iee] *Official IEEE 802.11 Working Group Project Timelines*. URL: http://grouper.ieee.org/groups/802/11/Reports/802.11_Timelines.htm.

[ISI81] University of Southern California Information Sciences Institute. *Transmission Control Protocol*. RFC 793. IETF, 1981.

[ITN13] O. Iova, F. Theoleyre, and T. Noel. "Stability and Efficiency of RPL under Realistic Conditions in Wireless Sensor Networks". In: *PIMRC*. IEEE. 2013.

[Jia+09] M. Jiang, Z. Guo, F. Hong, Y. Ma, and H. Luo. "OceanSense: A Practical Wireless Sensor Network on the Surface of the Sea". In: *PerCom*. IEEE, 2009.

[Kad+05] I. Kadayif, M. Kandemir, N. Vijaykrishnan, and M.J. Irwin. "An Integer Linear Programming-based Tool for Wireless Sensor Networks". In: *Journal of Parallel and Distributed Computing, Elsevier* 65.3 (2005), pp. 247 –260.

[Kam+12] P.O. Kamgueu, E. Nataf, T. D. Ndie, and O. Festor. *Energy-Based Routing Metric for RPL*. Research Report RR-8208. INRIA, 2012. URL: http://hal.inria.fr/hal-00779519.

[Kan+12] J. Kang et al. "PEAKSAVE: Energy Monitoring Service". In: *SenSys*. ACM, 2012.

[KB06] C. E. Koksal and H. Balakrishnan. “Quality-Aware Routing Metrics for Time-Varying Wireless Mesh Networks”. In: *Journal on Selected Areas in Communications, IEEE* 24.11 (2006), pp. 1984–1994.

[KCA07] A. Koubaa, A. Cunha, and M. Alves. “A Time Division Beacon Scheduling Mechanism for IEEE 802.15.4/Zigbee Cluster-Tree Wireless Sensor Networks”. In: *ECRTS*. IEEE, 2007.

[KDB13] R. Kacimi, R. Dhaou, and A.-L. Beylot. “Load Balancing Techniques for Lifetime Maximizing in Wireless Sensor Networks”. In: *Ad Hoc Networks, Elsevier* 11.8 (2013), pp. 2172 –2186.

[Kes91] S. Keshav. “A Control-Theoretic Approach to Flow Control”. In: *SIGCOMM*. ACM, 1991.

[Kim+07] K. Kim, S. Daniel Park, G. Montenegro, S. Yoo, and N. Kushalnagar. *6LoWPAN Ad Hoc On-Demand Distance Vector Routing (LOAD) draft-daniel-6lowpan-load-adhoc-routing-03.txt*. Internet-Draft. IETF, 2007.

[Kim+10] J. Kim, H. Kim, J. Jeong, Y. Seo, and P. Mah. “Field Deployment of a Large-Scale WSN for Parking Management System”. In: *SECON*. IEEE, 2010.

[KK00] B. Karp and H. T. Kung. “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks”. In: *MobiCom*. ACM, 2000.

[Ko+11] JG. Ko, S. Dawson-Haggerty, O. Gnawali, D. Culler, and A. Terzis. “Evaluating the Performance of RPL and 6LowPAN in TinyOS”. In: *IPSN*. ACM, 2011.

[Ko+14] J. Ko, J. Jeong, J. Park, J. Jun, N. Kim, and O. Gnawali. *RPL Routing Pathology In a Network With a Mix of Nodes Operating in Storing and Non-Storing Modes draft-ko-roll-mix-network-pathology-04*. Internet-Draft. IETF, 2014.

[Kou+06] A. Koubaa, B. Nefzi, M. Alves, and Song Y.-Q. *Improving the IEEE 802.15.4 Slotted CSMA/CA MAC for Time-Critical Events in Wireless Sensor Networks*. Technical Report. Polytechnic Institute of Porto, 2006.

[KS01] S. Kumar and D. Shepherd. “SensIT: Sensor Information Technology for the Warfighter”. In: *Proceedings of the 4th International Society of Information Fusion Conference*. 2001.

[Kuh+10] F. Kuhn, N. Lynch, C. Newport, R. Oshman, and A. Richa. “Broadcasting in unreliable radio networks”. In: *PODC*. ACM. 2010.

[KW10] D. Katz and D. Ward. *Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)*. RFC 5881. IETF, 2010.

[LBV06] K. Langendoen, A Baggio, and O. Visser. “Murphy Loves Potatoes: Experiences from a Pilot Sensor Network Deployment in Precision Agriculture”. In: *IPDPS*. IEEE, 2006.

[Lee+11] W.-Y. Lee, W.-I. Hwang, Y.-A. Jeon, and S.-S. Choi. “Distributed Fast Beacon Scheduling for Mesh Networks”. In: *MASS*. IEEE. 2011.

[Lee+14] K. Lee, H.-J. Kim, S. Park, and S. Han. “Satisfying the Target Network Lifetime in Wireless Sensor Networks”. In: *Computer Networks, Elsevier* 65.0 (2014), pp. 41 –55.

[Li+10] S. Li, R. K. Neelisetti, C. Liu, and A. Lim. “Efficient Multi-Path Protocol for Wireless Sensor Networks”. In: *International Journal of Wireless and Mobile Networks* 2.1 (2010), pp. 110–130.

[Liu+09] T. Liu, A Kamthe, L. Jiang, and A. Cerpa. “Performance Evaluation of Link Quality Estimation Metrics for Static Multihop Wireless Sensor Networks”. In: *SECON*. 2009.

[LK01] H. Lim and C. Kim. "Flooding in wireless ad hoc networks". In: *Computer Communications, Elsevier* 24.3-4 (2001), pp. 353–363.

[LL07] M. Li and Y. Liu. "Underground Structure Monitoring with Wireless Sensor Networks". In: *IPSN*. ACM, 2007.

[LSF07] H. Liu, J. Srivastava, and V. Firoiu. "PWave: A Multi-source Multi-sink Anycast Routing Framework for Wireless Sensor Networks". In: *Networking*. IFIP. Springer, 2007.

[LW08] J. Lee and J.C. Walrand. *Design and Analysis of an Asynchronous Zero Collision MAC Protocol*. arXiv:0806.3542. 2008.

[Mao+12] X. Mao, X. Miao, Y. He, X.-Y.i L, and Y. Liu. "CitySee: Urban CO2 Monitoring with Sensors". In: *INFOCOM*. IEEE, 2012.

[Mar+10] J. Martocci, P. De Mil, N. Riou, and W. Vermeylen. *Building Automation Routing Requirements in Low-Power and Lossy Networks*. RFC 5867. IETF, 2010.

[MhRlXd11] T. Ming-hao, Y. Ren-lai, and W. Xiang-dong. "Multipath Routing Protocol with Load Balancing in WSN Considering Interference". In: *ICIEA*. IEEE. 2011.

[Mic] *MICA2DOT, produced by Crossbow*. URL: `http://www.willow.co.uk/html/mpr5x0-_mica2dot_series.php`.

[Mo+09] L. Mo, Y. He, Y. Liu, J. Zhao, S.-J. Tang, X.-Y. Li, and G. Dai. "Canopy Closure Estimates with GreenOrbs: Sustainable Sensing in the Forest". In: *SenSys*. ACM, 2009.

[Mon+07] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*. RFC 4944. IETF, 2007.

[Nar+07] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. *Neighbor Discovery for IP version 6 (IPv6)*. RFC 4861. IETF, 2007.

[Ns2] *The Network Simulator - NS2*. URL: `http://www.isi.edu/nsnam/ns/`.

[O'F+07] B. O'Flyrm, R. Martinez, J. Cleary, C. Slater, F. Regan, D. Diamond, and H. Murphy. "SmartCoast: A Wireless Sensor Network for Water Quality Monitoring". In: *LCN*. IEEE, 2007.

[Ope] *OpenWSN*. https://openwsn.atlassian.net.

[Pax96] V. Paxson. "End-to-end routing behavior in the Internet". In: *ACM SIGCOMM*. 1996.

[PBRD03] C. Perkins, E. Belding-Royer, and S. Das. *Ad hoc On-Demand Distance Vector (AODV) Routing*. RFC 3561. IETF, 2003.

[PHC04] J. Polastre, J. Hill, and D. Culler. "Versatile Low Power Media Access for Wireless Sensor Networks". In: *SenSys*. ACM, 2004.

[PHT12] B. Pavkovic, W.-J. Hwang, and F. Theoleyre. "Cluster-Directed Acyclic Graph Formation for IEEE 80215.4 in Multihop Topologies". In: *NTMS*. IFIP/IEEE. 2012.

[Pis+09] K. Pister, P. Thubert, S. Dwars, and T. Phinney. *Industrial Routing Requirements in Low-Power and Lossy Networks*. RFC 5673. IETF, 2009.

[Pos80] J. Postel. *User Datagram Protocol*. RFC 768. IETF, 1980.

[PPR13] P. Pinto, A Pinto, and M. Ricardo. "End-to-end Delay Estimation Using RPL Metrics in WSN". In: *Wireless Days*. IFIP. 2013.

[PSC05] J. Polastre, R. Szewczyk, and D. Culler. "Telos: Enabling Ultra-low Power Wireless Research". In: *IPSN*. 2005.

[PSH06] J. Pak, J. Son, and K. Han. "A MAC Protocol Using Separate Wakeup Slots for Sensor Network". In: *Computational Science and Its Applications, Springer.* Vol. 3981. 2006, pp. 1159–1168.

[PTD11] B. Pavković, F. Theoleyre, and A. Duda. "Multipath Opportunistic RPL Routing over IEEE 802.15.4". In: *MSWiM.* ACM. 2011.

[Rad+12] M. Radi, B. Dezfouli, K.A. Bakar, and M. Lee. "Multipath Routing in Wireless Sensor Networks: Survey and Research Challenges". In: *Sensors* 12.1 (2012), pp. 650–685.

[RI04] V. Raisinghani and S. Iyer. "Cross-layer Design Optimizations in Wireless Protocol Stacks". In: *Computer Communications* 27.8 (2004), pp. 720 –724.

[Rio] *RIOT: The friendly Operating System for the Internet of Things.* http://riot-os.org/.

[RM12] E. Rescorla and N. Modadugu. *Datagram Transport Layer Security Version 1.2.* RFC 6347. IETF, 2012.

[ROGLA06] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. "Energy-efficient, Collision-free Medium Access Control for Wireless Sensor Network". In: *Wireless Networks, ACM* 12.1 (2006), pp. 63–78.

[SAZ10] Lifeng Sang, Anish Arora, and Hongwei Zhang. "On link asymmetry and one-way estimation in wireless sensor networks". In: *Transactions on Sensor Networks, ACM* 6.2 (2010), pp. 1–25.

[Sfp] 2014. URL: http://www.sfpark.org.

[SHB14] Z. Shelby, K. Hartke, and C. Bormann. *The Constrained Application Protocol (CoAP).* RFC 7252. IETF, 2014.

[Sim+04] G. Simon, M. Maróti, A. Lédeczi, G. Balogh, B. Kusy, A. N., G. Pap, J. Sallai, and K. Frampton. "Sensor Network-Based Countersniper System". In: *SenSys.* ACM, 2004.

[SL06] K. Srinivasan and P. Levis. "RSSI is Under Appreciated". In: *EmNets.* ACM, 2006.

[Sma] *Smart Gains.* URL: http://www.smartgrains.com.

[Sob03] J.L. Sobrinho. "Network Routing with Path Vector Protocols: Theory and Applications". In: *SIGCOMM.* ACM, 2003.

[Soh+00] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie. "Protocols for Self-organization of a Wireless Sensor Network". In: *Personal Communications, IEEE* 7.5 (2000), pp. 16–27.

[SS01] C. Schurgers and M.B. Srivastava. "Energy efficient routing in wireless sensor networks". In: *MILCOM.* IEEE. 2001.

[Tal+07] I. Talzi, A. Hasler, S. Gruber, and C. Tschudin. "PermaSense: Investigating Permafrost with a WSN in the Swiss Alps". In: *EmNets.* ACM, 2007.

[Thurc] P. Thubert. *Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL).* RFC 6552. IETF, March 2012.

[Tia+13] B. Tian, K.M. Hou, H. Shi, X. Liu, X. Diao, J. Li, Chen Y., and J.-P. Chanet. "Application of Modified RPL Under VANET-WSN Communication Architecture". In: *ICCIS.* 2013.

[Tin] *TinyOS.* http://www.tinyos.net/.

[TM06] J. Tsai and T. Moors. "A Review of Multipath Routing Protocols". In: *ACoRN Early Career Researcher Workshop on Wireless Multihop Networking.* 2006.

[Tmo] *TMote Sky, produced by Moteiv.*

[Tol+05] G. Tolle et al. “A Macroscope in the Redwoods”. In: *SenSys*. ACM, 2005.

[Tri] *Trilliant Communications Platform*. URL: `http://trilliantinc.com/`.

[Tri+10] J. Tripathi et al. “A performance evaluation study of RPL: Routing Protocol for Low power and Lossy Networks”. In: *CISS*. IEEE. 2010.

[Tri12] Tripathi, J. and Oliveira, de J. and Vasseur, JP. *Performance Evaluation of the Routing Protocol for Low-Power and Lossy Networks (RPL)*. RFC 6687. IETF, 2012. URL: `http://tools.ietf.org/html/rfc6687`.

[Tri13] Tripathi, J. and Oliveira, de J. “On Adaptive Timers for Improved RPL Operation in Low-power and Lossy Sensor Networks”. In: *COMSNETS*. IEEE, 2013.

[Var01] A. Varga. “The OMNET++ discrete event simulation system”. In: *ESM*. 2001.

[Vas+05] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke. “Data Collection, Storage, and Retrieval with an Underwater Sensor Network”. In: *SenSys*. ACM, 2005.

[Vas+12] JP. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel. *Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks*. RFC 6551. IETF, 2012.

[Ver+08] R. Verdone, D. Dardari, G. Mazzini, and A. Conti. “Wireless Sensor and Actuator Networks”. In: *Technologies, Analysis and Design*. Academic Press, 2008, pp. 1–11.

[VTD13] M. Vucinic, B. Tourancheau, and A Duda. “Performance Comparison of the RPL and LOADng Routing Protocols in a Home Automation Scenario”. In: *WCNC*. IEEE. 2013.

[WADHW08] G. Werner-Allen, S. Dawson-Haggerty, and M. Welsh. “Lance: Optimizing High-Resolution Signal Collection in Wireless Sensor Networks”. In: *SenSys*. ACM, 2008.

[Wan+10] D. Wang, Z. Tao, Zhang. J., and AA Abouzeid. “RPL Based Routing for Advanced Metering Infrastructure in Smart Grid”. In: *ICC*. IEEE. 2010.

[War+01] B. Warneke, M. Last, B. Liebowitz, and K. S. J. Pister. “Smart Dust: Communicating with a Cubic-millimeter Computer”. In: *Computer* 34.1 (2001), pp. 44–51.

[War+02] B. Warneke, M.D. Scott, B.S. Leibowitz, Lixia Zhou, C.L. Bellew, J.A. Chediak, J.M. Kahn, B.E. Boser, and K.S.J. Pister. “An Autonomous 16 mm^3 Solar-powered Node for Distributed Wireless Sensor Networks”. In: *Proceedings of Sensors*. IEEE, 2002.

[Wat+11] T. Watteyne, A Molinaro, M. G. Richichi, and M. Dohler. “From MANET to IETF ROLL Standardization: A Paradigm Shift in WSN Routing Protocols”. In: *Communications Surveys Tutorials, IEEE* 13.4 (2011), pp. 688–707.

[Whi05] E. C. Whitman. “SOSUS The "Secret Weapon" of Undersea Surveillance”. In: *Undersea Warfare* 7.2 (2005).

[Win+12] T. Winter et al. *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*. RFC 6550. IETF, 2012.

[WL12] F. Wang and J. Liu. “On Reliable Broadcast in Low Duty-Cycle Wireless Sensor Networks”. In: *Transactions on Mobile Computing, IEEE* 11.5 (2012), pp. 767 –779.

[Won12] C.-M. Wong. “A Novel Beacon Frame Scheduling Algorithm Based on Cluster-tree IEEE 802.15.4 Wireless Sensor Networks”. In: *ICCS*. IEEE. 2012.

[Wsn] *WSNet / Worldsens simulator*. `http://wsnet.gforge.inria.fr/`.

[WVW14] Q. Wang, X. Vilajosana, and T. Watteyne. *6TiSCH Operation Sublayer (6top) draft-wang-6tisch-6top-sublayer-01*. Internet-Draft. IETF, 2014.

[Xie+10] W. Xie, M. Goyal, H. Hosseini, J. Martocci, Y. Bashir, E. Baccelli, and A Durresi. "Routing Loops in DAG-Based Low Power and Lossy Networks". In: *AINA*. IEEE. 2010.

[XL13] G. Xu and G. Lu. "Multipath Routing Protocol for DAG-based WSNs with Mobile Sinks". In: *ICCSEE*. Atlantis Press, 2013.

[YBo09] B Yahya and J. Ben-othman. "REER: Robust and Energy Efficient Multipath Routing Protocol for Wireless Sensor Networks". In: *GLOBECOM*. IEEE. 2009.

[YHE02] W. Ye, J. Heidemann, and D. Estrin. "An energy-efficient MAC protocol for wireless sensor networks". In: *INFOCOM*. IEEE, 2002.

[YMG08] J. Yick, B. Mukherjee, and D. Ghosal. "Wireless Sensor Network Survey". In: *Computer Networks, Elsevier* 52.12 (2008), pp. 2292–2330.

[Yoo+10] H. Yoo, M. Shim, D. Kim, and K.H. Kim. "GLOBAL: A Gradient-based Routing Protocol for Load-balancing in Large-scale Wireless Sensor Networks with Multiple Sinks". In: *ISCC*. IEEE. 2010.

[Zha+04] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi. "Hardware Design Experiences in ZebraNet". In: *SenSys*. ACM, 2004.

List of Figures

List of Tables

List of Abbreviations

6TOP	6TSCH Operation Sublayer.
6TiSCH	IPv6 over the TSCH mode of IEEE 802.15.4e.
AMI	Advanced Metering Infrastructure.
AODV	Ad hoc On-Demand Distance Vector.
ARPANET	Advanced Research Projects Agency Network.
BFD	Bidirectional Forwarding Detection.
BI	Beacon Interval.
BO	Beacon Order.
BOP	Beacon Only Period.
BSN	Broadcast Sequence Number.
CAP	Contention Access Period.
CBOP	Contention Broadcast Only Period.
CBR	Constant Bit Rate.
CCDF	Complementary Cumulative Distribution Function.
CoAP	Constrained Application Protocol.
CoRE	Constrained RESTful Environments.
CPU	Central Processing Unit.
CSMA	Carrier Sense Multiple Access.
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance.
CTP	Collection Tree Protocol.
DAG	Directed Acyclic Graph.
DAO	Destination Advertisement Object.
DARPA	Defense Advanced Research Projects Agency.
DIO	DODAG Information Object.
DIS	DODAG Information Solicitation.
DODAG	Destination-Oriented Directed Acyclic Graph.
DRF	Dynamic Random Flooding.
DSN	Distributed Sensor Networks.
DTLS	Datagram Transport Layer Security.
ED	Energy Detection.
ELT	Expected Lifetime.
ENT	Effective Number of Transmissions.
ETT	Expected Transmission Time.
ETX	Expected Transmission Count.

EWMA	Exponential Weighted Moving Average.
FDMA	Frequency Division Multiple Access.
FFD	Full Function Device.
GBR	Gradient Based Routing.
GPS	Global Positioning System.
GPSR	Greedy Perimeter Stateless Routing.
GTS	Guaranteed Time Slot.
IBS	Inter-Beacon Space.
ICMPv6	Internet Control Message Protocol version 6.
IEEE	Institute of Electrical and Electronics Engineers.
IETF	Internet Engineering Task Force.
ISM	Industrial, Scientific and Medical radio bands.
LLN	Low-power and Lossy Network.
LOAD	6LoWPAN Ad Hoc On-Demand Distance Vector.
LOADng	Lightweight On-demand Ad hoc Distance-vector Routing Protocol - Next Generation.
LQI	Link Quality Indicator.
MAC	Medium Access Control.
MOP	Mode of Operation.
MRHOF	Minimum Rank with Hysteresis Objective Function.
NSA	Node State and Attribute.
NUD	Neighbor Unreachability Detection.
OF0	Objective Function Zero.
ORPL	Opportunistic RPL.
OSI	Open Systems Interconnection.
PAN	Personal Area Network.
PC	Personal Computer.
PDA	Personal Digital Assistant.
PDR	Packet Delivery Ratio.
PLC	Power-Line Communication.
PRR	Packet Reception Ratio.
QoS	Quality of Service.
RFD	Reduce Function Device.
RNP	Required Number of Packets.
ROLL	Routing Over Low power and Lossy networks.
RPL	IPv6 Routing Protocol for Low-power and Lossy Networks.
RSSI	Received Signal Strength Indicator.
RTS/CTS	Request to Send/Clear to Send.
RTT	Per-hop Round Trip Time.
SD	Superframe Duration.
SensIT	Sensor Information Technology.
SNR	Signal-to-Noise Ratio.

SO	Superframe Order.
SOSUS	Sound Surveillance System.
TCP	Transmission Control Protocol.
TDMA	Time Division Multiple Access.
TSCH	Timeslotted Channel Hopping.
UDP	User Datagram Protocol.
VANET	Vehicular Ad Hoc Network.
WCETT	Weighted Cumulative Expected Transmission Time.
WSN	Wireless Sensor Network.

Optimisation de standards et maximisation du temps de vie d'un réseau de capteurs pour l'Internet des objets

Introduction

L'Internet des Objets s'est développé récemment, permettant la communication directe entre des objets intelligents. L'IETF et l'IEEE ont standardisé des nouveaux protocoles spécifiques pour ce type de réseaux: RPL (pour la couche routage) et IEEE 802.15.4 (pour l'accès au médium).

RPL est un protocole à vecteur de distance, qui construit un graphe acyclique orienté vers la destination (DODAG), enraciné dans le routeur de bordure. La topologie est construite et maintenu par la diffusion de messages DIO (DODAG Information Objet). La fréquence d'envoi des DIOs est ajustée de manière dynamique, en utilisant l'algorithme trickle timer: afin d'économiser de l'énergie, plus la topologie est stable, plus la fréquence d'envoi des messages de contrôle diminue. Quand une incohérence est détectée, trickle réinitialise son temporisateur et les nœuds envoient des DIOs plus fréquemment, afin de propager rapidement la mise à jour du DODAG.

Chaque nœud choisit son parent préféré (le prochain saut) en fonction d'un ensemble de métriques de routage (le nombre de sauts, l'indicateur de qualité de lien, etc.). Une bonne métrique doit par conséquent tenir compte de la qualité des liens radio constituant le chemin vers la racine du DODAG. Bien que la métrique doive réagir rapidement aux changements, elle doit aussi être suffisamment stable pour éviter les reconfigurations inutiles du DODAG.

Les principales contributions de cette thèse sont:

1. Stabilité et efficacité de RPL
2. Diffusion fiable pour IEEE 802.15.4
3. Maximisation de la durée de vie du réseau
4. RPL multiparent

Stabilité et efficacité de RPL

Nous avons d'abord étudié l'efficacité du protocole de routage RPL. Pour ce faire, nous avons analysé le comportement de RPL dans le simulateur WSNet, utilisant un modèle de propagation radio réaliste calibré pour un scénario de type *indoor*. Pour étudier la stabilité et l'efficacité du RPL, nous avons implémenté plusieurs métriques de routage pour la construction du DODAG: MinHop (le nombre de sauts entre une source et la destination), ETX (le nombre attendu de transmissions avant qu'un paquet soit correctement acquitté par la destination) et LQI (un estimateur de la qualité du lien fourni par la puce radio).

Au travers de simulations, nous avons évalué de façon approfondie le comportement de ces métriques de routage. Nous avons notamment montré qu'aucune métrique ne réussit à garantir l'efficacité de RPL. En effet, chacune d'entre elles propose un compromis entre stabilité et efficacité énergétique. Par exemple, la métrique MinHop conduit à un DODAG très stable, mais avec de mauvaises performances. En effet, MinHop a tendance à sélectionner de mauvais liens radio. La métrique LQI est très stable et limite les variations du DODAG, mais offre de plus grands délais de bout en bout. À l'opposé, ETX réussit à équilibrer plus efficacement la charge entre les différents nœuds, mais génère une surcharge importante à cause des reconfigurations du DODAG.

Nous avons montré que les instabilités de RPL étaient préjudiciables à la vie du réseau : quelle que soit la métrique de routage, un nœud change son prochain saut. Ces oscillations génèrent une surcharge importante en raison de l'algorithme trickle: lorsqu'une incohérence est détectée, le temporisateur est remis à zéro et les paquets de contrôle sont transmis plus fréquemment. Corolairement, un nœud gaspille donc de l'énergie.

Diffusion fiable pour IEEE 802.15.4

La construction et la maintenance du DODAG se fait par la diffusion de messages DIO. Il est de la plus haute importance que la couche MAC utilise un mécanisme de diffusion fiable, sans quoi le protocole de routage se basera sur des informations incohérentes. C'est pourquoi nous avons conçu une nouvelle méthode de diffusion de la couche MAC dans la norme IEEE 802.15.4.

Dans IEEE 802.15.4 en mode balise, la communication entre les noeuds est synchronisé à l'aide des supertrames (périodes de temps durant lesquelles les noeuds sont altenartivement actifs puis endormis). Les périodes d'activité doivent être soigneusement planifiées pour éviter les collisions entre deux balises ou les paquets de données.

Nous avons proposé qu'un coordinateur choisisse aléatoirement une valeur Inter-Balise (IBS), constante pour toutes ses balises. Puis, quand la supertrame débute, il faudra patienter pendant cette valeur aléatoire avant d'envoyer sa trame baliser. De cette façon, nous prenons en considération efficacement les balises de taille variable. Nous avons montré par simulation et à travers une analyse probabiliste que cette méthode réduit le gaspillage de la bande passante.

Nous avons ensuite propose qu'un coordinateur envoie ses paquets de diffusion juste après l'envoi de sa trame balise. Nous évitons ainsi les collisions (un autre coordinateur doit vérifier que le médium radio est libre avant de transmettre ses propres balises). Il maintient également un numéro de séquence de diffusion (BSN) qui est ajouté dans toutes ses balises: chaque fois que le coordinateur doit envoyer un paquet en diffusion à ses voisins, il incrémente la valeur du BSN et il transmet également le paquet de diffusion correspondant. Un noeud peut demander les paquets de diffusion manquant en comparant le BSN dans la balise avec celui enregistré dans sa table de voisinage.

Maximisation de la durée de vie du réseau

Après avoir montré clairement qu'aucune métrique de routage existante ne réussit à garantir l'efficacité de RPL, nous avons défini la métrique durée de vie attendue (ELT). Elle possède les propriétés suivantes:

- elle capture les variations de la qualité du lien (de façon dynamique);
- elle maximise la fiabilité;
- elle minimise la consommation d'énergie des goulots d'étranglement (c.à.d., les nœuds qui consomment le plus d'énergie). Équilibrer la consommation d'énergie est primordial afin de prolonger la durée de vie du réseau.

Un nœud sélectionne ensuite son parent préféré tel que celui-ci maximise la durée de vie du nœud le plus contraint du réseau.

Nous avons enfin comparé notre métrique avec celles déjà existantes, à l'aide du simulateur WSNet. Nous avons remarqué que l'utilisation de la métrique ELT avec le protocole RPL parvient à égaler la fiabilité offerte par ETX, tout en réduisant la consommation énergétique maximale. En particulier, la charge est distribuée proportionnellement à la quantité d'énergie, de trafic et la qualité de chacun des nœuds vers leur parent préféré.

RPL multiparent

RPL construit dans un mode distribué un graphe acyclique orienté (DAG) dont la racine est le routeur de bordure. Cependant, il exploite un unique parent (le parent préféré) pour transmettre tous ses paquets. Nous avons proposé d'exploiter dans RPL plusieurs parents pour réduire l'instabilité induite par certains des métriques de routage.

Dans notre RPL multiparent, nous faisons une distinction claire entre le plan de contrôle et le plan de données. Nous modifions le concept de parent préféré: il représente uniquement le parent utilisé pour construire une structure de routage sans boucle (plan de contrôle).

Dans le plan de données, un nœud relaie ensuite ses paquets à tous ses parents, en prenant soin d'équilibrer leur consommation énergétique. Plus précisément, un nœud doit équilibrer de façon distribuée la charge des goulots d'étranglement identifiés dans toutes les routes. Nous avons également proposé un mécanisme de calcul de poids permettant de lisser la redirection de trafic dans le DODAG afin de rendre la topologie de routage stable. Nous améliorions ainsi à la fois la stabilité et l'équilibrage de la consommation d'énergie dans le réseau.

Des simulations avec un modèle de propagation réaliste ont permis de mettre en exergue l'intérêt d'une telle approche par rapport aux mécanismes de routage initialement proposés par RPL.

Oana-Teodora IOVA

Standards Optimization and Network Lifetime Maximization for Wireless Sensor Networks in the Internet of Things

Résumé

De nouveaux protocoles ont été standardisés afin d'intégrer les réseaux de capteurs sans fil (WSN) dans l'Internet. Parmi eux, RPL pour la couche routage et IEEE 802.15.4 pour la couche MAC. L'objectif de cette thèse est d'améliorer ces protocoles en prenant compte des contraintes énergétiques des dispositifs du WSN. Tout d'abord, nous avons conçu une nouvelle méthode de diffusion dans la norme IEEE 802.15.4, afin d'assurer une livraison fiable des paquets de contrôle des couches supérieures. Ensuite, nous avons fourni une évaluation exhaustive de RPL, en soulignant un problème d'instabilité qui génère une surcharge d'énergie importante. Compte tenu que la durée de vie des WSN est très limitée, nous avons aussi proposé une nouvelle métrique de routage qui identifie les goulets d'étranglement énergétiques afin de maximiser la durée de vie du réseau. Enfin, en couplant cette mesure avec une version multiparent de RPL, nous avons résolu le problème d'instabilité souligné précédemment.

Résumé en anglais

New protocols have been standardized in order to integrate Wireless Sensor Networks (WSN) in the Internet. Among them, the IEEE 802.15.4 MAC layer protocol, and RPL, the IPv6 Routing Protocol for Low-power and Lossy Networks. The goal of this thesis is to improve these protocols, considering the energy constraints of the devices that compose the WSN. First, we proposed a new MAC layer broadcast mechanism in IEEE 802.15.4, to ensure a reliable delivery of the control packets from the upper layers (especially from RPL). Then, we provided an exhaustive evaluation of RPL and highlighted an instability problem. This instability generates a large overhead, consuming a lot of energy. Since the lifetime of WSN is very limited, we proposed a new routing metric that identifies the energy bottlenecks and maximizes the lifetime of the network. Finally, by coupling this metric with a multipath version of RPL, we are able to solve the instability problem previously highlighted.